



ПАК ЯКОРЬ-ТТР-ТФ

РУКОВОДСТВО АДМИНИСТРАТОРА

(версия 1.0)

Авторские права

Без предварительного письменного разрешения, полученного от ООО «НЕОС», документ и любые выдержки из него, с изменениями и переводом на другие языки, не должны быть воспроизведены или использованы.

Содержание

1 ОБЩИЕ СВЕДЕНИЯ	5
1.1 НАЗНАЧЕНИЕ ДОКУМЕНТА	5
1.2 СОСТАВ ДОКУМЕНТА	5
1.3 ТЕХНИЧЕСКАЯ ПОДДЕРЖКА.....	6
1.3.1 Производитель	6
1.3.2 Служба технической поддержки	6
1.4 ИСПОЛЬЗУЕМЫЕ ТЕРМИНЫ И СОКРАЩЕНИЯ.....	7
1.5 ИСТОРИЯ ИЗМЕНЕНИЙ.....	8
2 ОПИСАНИЕ СИСТЕМЫ	9
2.1 АРХИТЕКТУРА СИСТЕМЫ.....	10
3 НАСТРОЙКА ПАРАМЕТРОВ КОНФИГУРАЦИИ	11
3.1 УСЛОВНЫЕ ОБОЗНАЧЕНИЯ.....	11
3.2 НАСТРОЙКА ПАРАМЕТРОВ ПРИЛОЖЕНИЯ LI.GW.86	13
3.2.1 Конфигурация основных параметров приложения (config.properties)	13
3.2.2 Конфигурация пультов управления ОРМ (ru.config.properties).....	26
3.3 НАСТРОЙКА ПАРАМЕТРОВ ПРИЛОЖЕНИЯ VOICE-STORAGE	27
3.3.1 Конфигурация параметров системы журналирования (trace.cfg)	27
3.3.2 Настройка основных параметров приложения (voice-storage.cfg)	34
3.4 НАСТРОЙКА ПАРАМЕТРОВ ПРИЛОЖЕНИЯ VOICE-PROBE	36
3.4.1 Настройка подсистемы Alarm Processor.....	36
3.4.2 Настройка основных параметров приложения (app.cfg).....	46
3.4.3 Настройка мониторинга физического уровня (monitor.cfg)	51
3.4.4 Настройка параметров взаимодействия (om_interface.cfg).....	54
3.4.5 Настройка параметров для приема данных потока E1 (tdm.cfg)	59
3.4.6 Настройка многопоточности (Threads.cfg)	62
3.4.7 Настройка параметров системы журналирования (trace.cfg)	62
3.4.8 Настройка параметров RTP-сессий (VOP.cfg).....	71
3.4.9 Настройка физических параметров аппаратного обеспечения PROBE для приема потоков E1	74
4 ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ	78
4.1 ДОСТУП С ИСПОЛЬЗОВАНИЕМ КОМАНДЫ SSH	78
4.2 УПРАВЛЕНИЕ РАБОТОЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	78
4.3 ПЕРЕЗАПУСК СИСТЕМЫ	79
4.4 ПРОВЕРКА РАБОТОСПОСОБНОСТИ	79
4.4.1 Проверка работы приложения	79
4.4.2 Проверка запуска приложения	80
4.4.3 Проверка состояния приложения VOICE-PROBE после запуска	81

4.4.4 Проверка состояния приложения LI.GW.86 после запуска	81
5 ЖУРНАЛЫ.....	82
5.1 ОБЩИЕ ПРИНЦИПЫ ОРГАНИЗАЦИИ ЖУРНАЛОВ	82
5.2 ОПИСАНИЕ ЖУРНАЛОВ ПРИЛОЖЕНИЯ LI.GW.86.....	82
5.2.1 Журнал работы приложения (access-point.log)	82
5.2.2 Журнал работы компонент приложения (profiler.log)	83
5.2.3 Журнал работы сенсоров (sensors.journal)	83
5.2.4 Журнал процесса работы сенсоров (sensors.log)	83
5.2.5 Журнал работы компоненты spring (spring.log).....	84
5.2.6 Системный журнал (system.log)	84
5.3 ОПИСАНИЕ ЖУРНАЛОВ ПРИЛОЖЕНИЯ VOICE-STORAGE.....	84
5.3.1 Базовые журналы приложения	84
5.3.2 Стандартный вывод и журнал скриптовой подсистемы	85
5.3.3 Журнал переданных RTP-поток (rtp_sender.log)	86
5.3.4 Журнал ошибок передачи RTP-поток (rtp_sender_warning.log).....	86
5.4 ОПИСАНИЕ ЖУРНАЛОВ ПРИЛОЖЕНИЯ VOICE-PROBE	86
5.4.1 Базовые журналы приложения	86
5.4.2 Журнал обработки сигнализации DSS1 (dss1_handler.log)	87
5.4.3 Журнал обработки сигнализации ISUP (isup_handler.log)	87
5.4.4 Журнал обработки голосового трафика (voice_manager.log).....	87
5.4.5 Журнал приема голосовой информации (voice_switch.log)	87
5.4.6 Журнал обработки голосовых данных из потока E1 (e1_2_rtp.log)	88
5.4.7 Журнал обработки перехватываемых вызовов (sorm_database.log)	88
5.4.8 Журнал загрузки приложения (initialize.log)	88
5.4.9 Журнал конфигурации аппаратного обеспечения (Ph_config.log)	88
5.4.10 Журнал состояния аппаратного обеспечения (Ph_info.log).....	88
5.4.11 Журнал работы аппаратного обеспечения (Ph_trace.log)	88
5.4.12 Журнал ошибок аппаратного обеспечения (Ph_warning.log).....	88

1 Общие сведения

1.1 Назначение документа

Настоящее руководство содержит руководство пользователя по работе с ПАК ЯКОРЬ-TTR-ТФ, разработки ООО «НЕОС».

1.2 Состав документа

Настоящее руководство состоит из следующих основных частей:

«Общие сведения» — раздел, описывающий назначение и состав документа, содержащий сведения о производителе и технической поддержке;

«Описание системы» — раздел, содержащий сведения о системе, ее характеристики и архитектуру системы;

«Настройка параметров конфигурации» — раздел, содержащий информацию о настройке основных параметров системы с помощью конфигурационных файлов;

«Техническое обслуживание» — раздел, содержащий описание работы пользователя с программным обеспечением;

«Журналы» — раздел, содержащий информацию о журналах системы.

Внимание!

Перед установкой и началом эксплуатации изделия необходимо внимательно ознакомиться с паспортом изделия и эксплуатационной документацией.

Данный документ должен постоянно находиться при изделии.

1.3 Техническая поддержка

Техническая поддержка, а также дополнительное консультирование по вопросам, возникающим в процессе эксплуатации приложения, осуществляются производителем и службой технической поддержки.

1.3.1 Производитель

ООО «НЕОС»

197342, Санкт-Петербург

ул. Белоостровская, д. 6, литера А, офис №55

Тел.: (812) 779-13-79

WEB: <https://www.neo-s.ru>

E-mail: sales@neo-s.com

1.3.2 Служба технической поддержки

ООО «НЕОС»

197342, Санкт-Петербург

ул. Белоостровская, д. 6, литера А, офис №55

Тел.: (812) 779-13-79 доп. 1 (круглосуточно)

WEB: <https://www.neo-s.ru>

E-mail: support@neo-s.com

1.4 Используемые термины и сокращения

Используемые в настоящем документе термины и сокращения приведены в таблице 1.

Таблица 1 — Используемые термины и сокращения

Термин/Сокращение	Расшифровка
БД	База данных.
ОКС-7	(Общий Канал Сигнализации №7) Набор сигнальных телефонных протоколов, используемых для настройки большинства телефонных станций.
ОРМ	Оперативно-розыскные мероприятия.
ПО	Программное обеспечение.
ПК	Программный комплекс
ПУ	Пульт управления.
ТС	Технические средства.
2ВСК	Сигнализация по двум выделенным сигнальным каналам.
E1	Цифровой поток передачи данных скоростью 2048 Кбит/с с тридцатью каналами для передачи голоса или данных и двумя каналами для сигнализации.
EDSS-1	(European Digital Subscriber Signaling System № 1) Европейская цифровая система сигнализации № 1 в цифровой сети ISDN.
ISUP	(ISDN User Part) Пользовательская часть (подсистема) ISDN.
RTP	(Real Time Transport Protocol) протокол передачи данных в реальном времени.
SIP	(Session Initiation Protocol) протокол инициирования сеансов связи.
SSH	(Secure Shell) сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов).
TCP/IP	(Transmission Control Protocol/Internet Protocol) протокол управления передачей/интернет протокол.
UDP	(User Datagram Protocol) протокол передачи датаграмм пользователей.

1.5 История изменений

История изменений настоящего документа фиксируется в таблице 2.

Таблица 2 — История изменений

Дата	Версия документа	Версия продукта	Комментарий
01.03.2023	1.0	Версия ПО: 1.0	Первая версия документа

2 Описание системы

Программное обеспечение «ПАК ЯКОРЬ-TTR-ТФ» (далее — ПАК ЯКОРЬ-TTR-ТФ) производства ООО «НЕОС» предназначен для сбора и хранения голосовой информации на сети телефонной связи, в том числе информации, передаваемой абонентами при заказе и отмене ДВО, в соответствии с требованиями Приказа Минкомсвязи России от 26.02.2018 № 86.

ПАК ЯКОРЬ-TTR-ТФ подключается к сети оператора связи различными способами:

- через mirroring-port на коммутаторе;
- через оптические сплиттеры, которые устанавливаются в разрыв оптических каналов связи;
- через съёмники потоков E1.

От точек съёма трафик через сеть передачи данных попадает на сервер обработки голосовой информации. Сервер настроен в соответствии с подаваемой на него нагрузкой и хранит информацию о конфигурации сигнальных линков, соответствии СІС транк-групп сети ОКС № 7 и физических TSL потоков E1 на сети оператора.

На основании перехваченной информации из сигнальных каналов и конфигурации сигнальных линков ПАК ЯКОРЬ-TTR-ТФ привязывает статистическую информацию о вызове к содержимому вызова, которое передается в конкретном разговорном канале или в RTP-сессии.

Перехваченные записи и контент перемещаются в систему хранения данных, где хранятся 6 месяцев.

ПАК ЯКОРЬ-TTR-ТФ предназначен для выполнения следующих функциональных характеристик:

- осуществлять подключение к каналам связи STM-1/E1/Ethernet 1GbE и 10 GbE в пассивном режиме;
- осуществлять обработку сигнальной и голосовой информации, передаваемой по каналам связи;
- осуществлять разбор протоколов сигнализации SIP/SIP-T/SIP-I, H.323, ОКС-7, EDSS-1, QSIG, TN1R6, CAS2;
- осуществлять декодирование речевого трафика RTP, сжатого речевыми кодеками G.711 (a-law и μ -law), G.729, G.722, G.723, G.726, AMR, AMR-WB;
- осуществлять запись вызовов и сохранение их в СХД;
- предоставлять доступ к сохраненным вызовам и статистической информации со стороны ПУ ОРМ в соответствии с требованиями приказа Минкомсвязи № 86 от 26.02.2018;
- осуществлять обработку сигнальной информации и формирование записей об использовании дополнительных видов обслуживания (ДВО);
- выполнять запись и сохранение в СХД обмена короткими сообщениями между абонентами;
- осуществлять обработку DTMF-сигналов;
- осуществлять управления с помощью интерфейса командной строки;
- выполнять сопряжение со стандартными средствами мониторинга (например, Zabbix);
- перехват и анализ трафика TETRA.

2.1 Архитектура системы

Система может использоваться на сети оператора связи.

ПАК ЯКОРЬ-TTR-ТФ имеет модульную структуру и состоит из следующих программных приложений: VOICE-PROBE, Voice-storage, LI.GW.86.

На рисунке ниже приведена схема взаимодействия между приложениями ПАК ЯКОРЬ-TTR-ТФ.

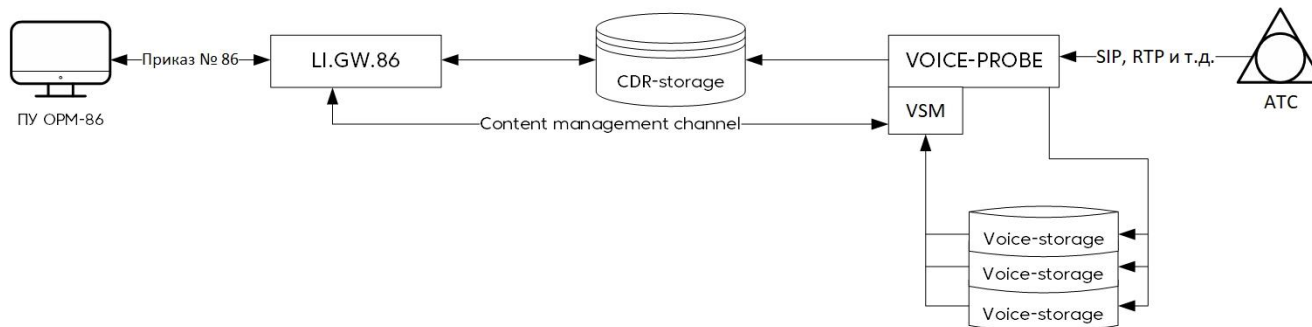


Рисунок 1 — Внутренняя архитектура ПАК ЯКОРЬ-TTR-ТФ

Основные элементы ПАК ЯКОРЬ-TTR-ТФ:

- LI.GW.86 — точка доступа для ПУ — приложение, реализующее интерфейс в сторону ПУ и обеспечивающее выполнение команд обработки данных через формирование SQL-запросов к CDR-storage и запросов контента к Voice-storage;
- CDR-storage — база данных, обеспечивающая хранение статистической информации о вызовах и возможность выборки вызовов по некоторым фильтрам;
- VOICE-PROBE — приложение съёма SIP и RTP-трафика, а также ряда других протоколов, которые требуются для реализации требований приказа;
- Voice-storage — приложение, которое получает от VOICE-PROBE RTP-трафик и обеспечивает его запись на диск, а также доступ к контенту;
- VSM или voice-storage-mux — утилита, которая выступает единой точкой входа для нескольких Voice-storage.

3 Настройка параметров конфигурации

Настройки управления параметрами конфигурации каждого из модулей системы сохраняются в файлах конфигурации в своей папке:

1. Конфигурация приложения LI.GW.86: /usr/neos/LI.GW.86/config;
2. Конфигурация приложения Voice-storage: /usr/neos/SNGI/config;
3. Конфигурация приложения VOICE-PROBE: /usr/neos/PROBE/config.

В данных файлах представлены стандартные для любой конфигурации секции.

3.1 Условные обозначения

При описании конфигурационных файлов используются общепринятые типы данных и набор условных обозначений.

Определения, используемые при описании конфигурации:

Условные обозначения

Графа «Значимость параметра/перезапуск» в таблицах конфигурации содержит буквенные коды.

В таблице ниже описаны характеристики параметров, для которых применяются данные обозначения.

Таблица 3 — Буквенные коды

Тип	Описание
O	Optional. Опциональный параметр. Может отсутствовать в конфигурации, в таком случае используется значение по умолчанию.
M	Mandatory. Обязательный параметр. Его отсутствие не позволяет запустить систему, а после перезагрузки конфигурации отображается сообщение об ошибке.
P	Permanent. Параметр не переопределяется динамически, поскольку используется при запуске системы.
R	Reloadable. Параметр, значение которого можно переопределить без перезагрузки.
U	Unique. Параметр, имеющий уникальное в данном контексте значение.
X	Параметр зарезервирован. Использование запрещено.

Типы данных

В ходе взаимодействия с сервисом происходит обмен данными определенных типов.

В таблице ниже описаны типы данных, которые применяются во время работы с сервисом.

Таблица 4 — Используемые обозначения для типов данных

Тип	Описание
bool	Логический тип. Используется для задания флага. Принимает только значения 0 или 1, false или true соответственно.
CA (ComponentAdress)	Адрес компоненты. Строка, представляющая собой путь к программной компоненте, состоящий из перечисления узлов дерева компонент, разделенных точкой

Тип	Описание
datetime	Тип для задания даты и времени. Используемые сокращения: YY/YYYY — год, записанный двумя/четырьмя цифрами соответственно; MM — месяц, записанный двумя цифрами; DD — день, записанный двумя цифрами; hh — часы, записанные двумя цифрами; mm — минуты, записанные двумя цифрами; ss — секунды, записанные двумя цифрами; mss — миллисекунды, записанные тремя цифрами. Время задается в формате 24-часового дня.
digit_string	Строковый идентификатор, может содержать только цифры.
enum	Целое знаковое число, элемент перечисления.
float	Дробное число.
format_string	Строковый идентификатор определенного формата (должен указываться формат).
hex	Числовой тип. Задает целое число в формате шестнадцатеричного числа, записанного цифрами 0–9 и буквами A–F. Числу может предшествовать обозначение 0x. При отсутствии обозначения определяется как строка.
int	Числовой тип. Задает целое 32-битное число, записанное цифрами 0–9 и знаком минуса "-". Диапазон: от -2^{31} до $2^{31}-1$.
list	Список, содержит несколько значений одной типа или структуры.
object	Кортеж, содержит фиксированное количество параметров различных типов.
regex	Регулярное выражение. Шаблон (маска) номера – формат параметра, задается с помощью регулярных выражений.
string	Строковый тип. Может содержать буквы латинского алфавита, цифры 0–9, спецсимволы и знаки препинания.
templ_selector	Простой шаблон.
unsigned	Целое беззнаковое число

Выражения

Выражения вида: «<[имя_параметра:]тип>» необходимо заменять на значения соответствующего типа.

Сложный параметр — подсекция конфигурационного файла. Содержит набор параметров.

Примечание — Сотрудникам технической поддержки заказчика допускается настраивать только параметры, непосредственно отвечающие за работу заказанной услуги по согласованию с предприятием-изготовителем оборудования.

3.2 Настройка параметров приложения LI.GW.86

Приложение LI.GW.86 осуществляет предоставление интерфейсов для подключения ПУ.

Настройка приложения LI.GW.86 осуществляется в следующих конфигурационных файлах:

- `config.properties` — файл, содержащий основные настройки параметров приложения;
- `ru.config.properties` — файл, содержащий настройки параметров пультов управления ОРМ.

3.2.1 Конфигурация основных параметров приложения (`config.properties`)

Настройки управления конфигурацией приложения сохраняются в файле `config.properties`.

В таблице ниже описаны параметры конфигурационного файла.

Таблица 5 — Параметры `config.properties`

Параметр	Значимость параметра/ перезапуск	Описание
Параметры взаимодействия с пультом управления.		
<code>transfer_inactivity_timeout</code>	О/Р	Максимальное время простоя (в секундах). Тип — <code>int</code> . Значение по умолчанию: 300
<code>max_transfer_inactivity_count</code>	О/Р	Максимальное количество возможных простоев. Тип — <code>int</code> . Значение по умолчанию: 3
<code>disconnect_on_fail</code>	О/Р	Прерывать соединение при возникновении ошибки. Тип — <code>bool</code> . Возможные значения: 1 — прерывать (<code>true</code>); 0 — не прерывать (<code>false</code>). Значение по умолчанию: <code>false</code>
<code>main_management_channel_port</code>	О/Р	Порт канала управления для головного ПУ. Тип — <code>int</code> . Значение по умолчанию: 16118
<code>main_data_channel_port</code>	О/Р	Порт канала передачи данных для головного ПУ. Тип — <code>int</code> . Значение по умолчанию: 16117

Параметр	Значимость параметра/ перезапуск	Описание
channel_start_stop_timeout	О/Р	Таймаут открытия и закрытия каналов (в секундах). Тип — int. Значение по умолчанию: 10
message_queue_capacity	М/Р	Размер очереди сообщений. Тип — int.
frame_queue_size_warning_threshold	О/Р	Порог размера очереди кадров для отправки, при котором в журнале (log-файле) выводится предупреждение. Тип — int. Значение по умолчанию: 10000
max_supported_count_queries	М/Р	Максимальное количество запросов, поддерживаемое для данного ПУ. Тип — int.
vendor_number	М/Р	Уникальный номер производителя технических средств ОРМ. Тип — string.
pu_config_path	М/Р	Путь к файлу конфигурации ПУ. Тип — string.
regional_distribution	О/Р	Включение режима территориального распределения. Тип — bool. Возможные значения: 1 — включить (true); 0 — не включать (false). Значение по умолчанию: false

Параметр	Значимость параметра/ перезапуск	Описание
used_local_time	О/Р	Использование времени в локальном часовом поясе (вместо UTC) при отправке на ПУ. Тип — bool. Возможные значения: 1 — использовать (true); 0 — не использовать (false). Значение по умолчанию: false
Параметры настройки потоков приложения.		
executor.shutdown_timeout	О/Р	Время ожидания завершения потоков приложения (в секундах). Тип — int. Значение по умолчанию: 5
content_loading.parallelism	О/Р	Максимальное количество потоков для загрузки контента. Тип — int. Значение по умолчанию: 100
content_loading.min_threads_per_query	О/Р	Минимальное количество потоков, обрабатывающих запрос (один поток может обрабатывать несколько запросов). Тип — int. Значение по умолчанию: 5
Настройка параметров соединения с БД (Stat Storage).		
stat_db.url	М/Р	URL-адрес для подключения к БД со статистикой. Тип — string.
stat_db.schema	О/Р	Наименование схемы БД. Тип — string.
stat_db.username	М/Р	Имя пользователя для подключения к БД. Тип — string.

Параметр	Значимость параметра/ перезапуск	Описание
stat_db.password	М/Р	Пароль для доступа к БД. Тип — string.
stat_db.max_active	О/Р	Максимальное количество активных соединений к БД в наборе («пул») Соединения созданы и поддерживаются таким образом, чтобы не было необходимости пересоздавать их для новых запросов. Тип — int. Значение по умолчанию: 10
stat_db.min_idle	О/Р	Минимальное количество простаивающих соединений к БД в наборе («пул») Тип — int. Значение по умолчанию: 1
Настройка параметров соединения с БД, в которой хранятся записи голосовой информации (Voice Data Storage).		
voice_storage.host	М/Р	Адрес сервера с записями голосовой информации. Тип — string.
voice_storage.port	М/Р	Порт сервера с записями голосовой информации. Тип — int.
Настройка параметров соединения с системой мониторинга и отслеживания статусов разнообразных сервисов (Zabbix агент).		
zabbix_agent.connection_timeout	М/Р	Время ожидания подключения к Zabbix агенту (в миллисекундах). Тип — int.
zabbix_agent.response_timeout	М/Р	Время ожидания ответа от Zabbix агента (в миллисекундах). Тип — int.

Параметр	Значимость параметра/ перезапуск	Описание
<p>Настройка параметров технических средств (далее — ТС) и программных компонент.</p> <p>Свойства <code>component.hardware.[0-...]</code> и <code>component.software.[0-...]</code> являются необязательными, но программный компонент (свойство <code>component.software.[0-...]</code>) не может существовать отдельно от оборудования (свойство <code>component.hardware.[0-...]</code>).</p>		
<code>component.hardware.[0-...].id</code>	M/P	Идентификатор оборудования ТС ОРМ. Тип — int.
<code>component.hardware.[0-...].block_number</code>	M/P	Номер устройства ТС ОРМ. Тип — int.
<code>component.hardware.[0-...].name</code>	M/P	Наименование оборудования ТС ОРМ. Тип — string.
<code>component.hardware.[0-...].host</code>	M/P	IP-адрес ТС ОРМ. Тип — string.
<code>component.hardware.[0-...].zabbix_port</code>	O/P	Порт Zabbix агента. Тип — int. Значение по умолчанию: 10050
<code>component.software.[0-...].hardware_id</code>	M/P	Идентификатор оборудования ТС ОРМ, на котором находится программный компонент. Тип — int.
<code>component.software.[0-...].name</code>	M/P	Наименование программного компонента на оборудовании ТС ОРМ. Тип — string.
<code>component.software.[0-...].type</code>	O/P	Тип программного компонента. Тип — string. Возможные значения: <code>common</code> и <code>probe</code> . Значение по умолчанию: <code>common</code>
<code>component.software.[0-...].pid_file</code>	M/P	Абсолютный путь к PID файлу программного компонента. Тип — string.

Параметр	Значимость параметра/ перезапуск	Описание
component.software.[0-...].ethernet.check	О/Р	<p>Включить проверку Ethernet интерфейсов.</p> <p>Тип — bool.</p> <p>Возможные значения:</p> <p>1 — включить (true);</p> <p>0 — не включать (false).</p> <p>Значение по умолчанию: false</p> <p>Примечание — Только если указан тип программного компонента probe.</p>
component.software.[0-...].ethernet.interfaces	М/Р	<p>Названия Ethernet интерфейсов, задаются через запятую.</p> <p>Тип — string.</p> <p>Формат записи: eth0,eth1</p> <p>Примечание — Только если указан тип программного компонента probe.</p>
component.software.[0-...].e1.check	О/Р	<p>Включить проверку потоков E1.</p> <p>Тип — bool.</p> <p>Возможные значения:</p> <p>1 — включить (true);</p> <p>0 — не включать (false).</p> <p>Значение по умолчанию: false</p> <p>Примечание — Только если указан тип программного компонента probe.</p>
component.software.[0-...].e1.snmp.port	О/Р	<p>Порт SNMP.</p> <p>Тип — int.</p> <p>Значение по умолчанию: 161</p> <p>Примечание — Только если указан тип программного компонента probe.</p>

Параметр	Значимость параметра/ перезапуск	Описание
component.software.[0-...].e1.snmp.trunks	O/P	<p>MIB OID списка транков.</p> <p>Тип — string.</p> <p>Значение по умолчанию: 1.3.6.1.4.1.20873.100.1.2.1.4096</p> <p>Примечание — Только если указан тип программного компонента probe.</p>
component.software.[0-...].e1.snmp.trunks_state	O/P	<p>MIB OID OSTATE транков.</p> <p>Тип — string.</p> <p>Значение по умолчанию: 1.3.6.1.4.1.20873.100.1.2.1.4097</p> <p>Примечание — Только если указан тип программного компонента probe.</p>
Параметры состояния технических и программных средств		
state_metrics.net_speed_averaging_interval	O/P	<p>Интервал времени, за который высчитывается среднее значение скорости сетевых интерфейсов (в секундах).</p> <p>Тип — int.</p> <p>Значение по умолчанию: 1</p>
Параметры сервиса состояний (AlarmService)		
alarm_service.path	M/P	<p>Путь к директории, в которой сохраняются сообщения с состоянием.</p> <p>Тип — string.</p>
alarm_service.scheduler_resend_interval	O/P	<p>Временной интервал поиска, отправки и удаления (если необходимо) сообщений состояний.</p> <p>Тип — int.</p> <p>Значение по умолчанию: 1000</p>
alarm_service.sending_interval	M/P	<p>Временной интервал для повторной отправки сообщений с состояниями (в миллисекундах) — T1.</p> <p>Тип — int.</p>

Параметр	Значимость параметра/ перезапуск	Описание
alarm_service.ram_sending_interval	O/P	<p>Временной интервал для повторной отправки сообщений с состоянием о заполнении памяти (в миллисекундах).</p> <p>Тип — int.</p> <p>Примечание — Значение данного параметра изменять не следует.</p> <p>Значение по умолчанию: 60000</p>
Параметры для хранилища состояний		
alarm.state_storage.operation_timeout	O/P	<p>Максимальное время ожидания выполнения операции (сохранение, обновление, очистка информации о состоянии) в миллисекундах.</p> <p>Тип — int.</p> <p>Значение по умолчанию: 60000</p>
Параметры сенсоров		
sensor.state_path	M/P	<p>Путь к директории, в которой сохраняются состояния сенсоров.</p> <p>Тип — string.</p>
sensor.common.polling_interval	M/P	<p>Общий интервал отправки запроса состояния у сенсоров (в миллисекундах).</p> <p>Тип — int.</p>
Sensor.common.preload_attempt_count	O/P	<p>Количество попыток предварительной загрузки сенсоров.</p> <p>Тип — int.</p> <p>Значение по умолчанию: 3</p>
.sensor.keyboard.polling_interval	O/P	<p>Интервал отправки запроса состояния у сенсора клавиатуры (в миллисекундах).</p> <p>Тип — int.</p>

Параметр	Значимость параметра/ перезапуск	Описание
sensor.log_in_out.polling_interval	O/P	Интервал отправки запроса состояния у сенсора подключения/отключения к хосту (в миллисекундах). Тип — int.
sensor.failed_login_attempts.polling_interval	O/P	Интервал отправки запроса состояния у сенсора неуспешных попыток подключения к хосту (в миллисекундах). Тип — int.
sensor.chassis_intrusion.polling_interval	O/P	Интервал отправки запроса состояния у сенсора корпуса (в миллисекундах). Тип — int.
sensor.ram.polling_interval	O/P	Интервал отправки запроса состояния у сенсора памяти (в миллисекундах). Тип — int.
sensor.ram.averaging_interval	O/P	Интервал, с помощью которого вычисляется средняя скорость заполнения памяти (в миллисекундах). Тип — int. Значение по умолчанию: 60000
sensor.probes_state.polling_interval	O/P	Интервал отправки запроса состояния у сенсора состояния съемников (в миллисекундах). Тип — int.
sensor.probes_state.max_allowed_net_speed	O/P	Верхняя граница допустимой скорости Ethernet интерфейсов съемников в байт/сек. Тип — int. Значение по умолчанию: 13107200

Параметр	Значимость параметра/ перезапуск	Описание
sensor.probes_state.net_speed_averaging_interval	O/P	Интервал времени, за который высчитывается среднее значение скорости сетевых интерфейсов съемников (в секундах). Тип — int. Значение по умолчанию: 1
sensor.eth_connection.polling_interval	O/P	Интервал отправки запроса состояния у сенсора Ethernet соединения (в миллисекундах). Тип — int.
sensor.e1_connection.polling_interval	O/P	Интервал отправки запроса состояния у сенсора соединения E1 (в миллисекундах). Тип — int.
sensor.cpu_temp.polling_interval	O/P	Интервал отправки запроса состояния у сенсора температуры CPU (в миллисекундах). Тип — int.
sensor.cpu_temp.min_allowed	O/P	Нижняя граница допустимой температуры центрального процессора. Тип — int. Значение по умолчанию: 0
sensor.cpu_temp.max_allowed	O/P	Верхняя граница допустимой температуры центрального процессора. Тип — int. Значение по умолчанию: 95
sensor.fan_speed.polling_interval	O/P	Интервал отправки запроса состояния у сенсора скорости вращения вентилятора (в миллисекундах). Тип — int.

Параметр	Значимость параметра/ перезапуск	Описание
sensor.fan_speed.min_allowed	O/P	Нижняя граница допустимой скорости вращения вентилятора. Тип — int. Значение по умолчанию: 600
sensor.fan_speed.max_allowed	O/P	Верхняя граница допустимой скорости вращения вентилятора. Тип — int. Значение по умолчанию: 18975
sensor.power_supply.polling_interval	O/P	Интервал опроса состояния у сенсора напряжения питания (в миллисекундах). Тип — int.
sensor.power_supply.min_allowed	O/P	Нижняя граница допустимой величины напряжения. Тип — string. Значение по умолчанию: 11.4
sensor.power_supply.max_allowed	O/P	Верхняя граница допустимой величины напряжения. Тип — string. Значение по умолчанию: 12.6
tracing.enable	O/P	Включить распределенную трассировку. Тип — bool. Возможные значения: 1 — включить (true); 0 — не включать (false). Значение по умолчанию: false
tracing.service_name	O/P	Имя сервиса в OpenTracing. Тип — string.

Пример файла конфигурации:

```
Common
transfer_inactivity_timeout=300
max_transfer_inactivity_count=3
```

```
main_management_channel_port=16118
main_data_channel_port=16117
channel_start_stop_timeout=10
message_queue_capacity=200
frame_queue_size_warning_threshold=10000
max_supported_count_queries=100
vendor_number=1
pu_config_path=config/pu_config.properties
used_local_time=true

Executors

executor.shutdown_timeout=5

DB connections

Stat storage
stat_db.url=jdbc:postgresql://172.16.202.152:5432/jarovaya
stat_db.schema=public
stat_db.username=sorm
stat_db.password=elephant
stat_db.max_active=10
stat_db.min_idle=1

Voice storage

voice_storage.host=172.16.202.152
voice_storage.port=12345

Monitoring

zabbix_agent.connection_timeout=5000
zabbix_agent.response_timeout=30000

Hardware and software components
component.hardware.0.id=1
component.hardware.0.block_number=1
component.hardware.0.name=LI.GW/86
component.hardware.0.host=127.0.0.1
component.hardware.0.zabbix_port=10050
```



```
component.software.0.hardware_id=1
component.software.0.name=LI.GW/86
component.software.0.pid_file=/usr/neos/LI.GW.86/run/li-gw-86.pid
```

```
component.software.1.hardware_id=3
component.software.1.name=SNGI
component.software.1.pid_file=/usr/neos/SNGI/bin/pids/sngi.pid
```

```
component.hardware.1.id=2
component.hardware.1.block_number=2
component.hardware.1.name=PROBE
component.hardware.1.host=172.16.202.149
component.hardware.1.zabbix_port=10050
```

```
component.software.2.hardware_id=2
component.software.2.type=probe
component.software.2.e1.check=true
component.software.2.name=PROBE
component.software.2.pid_file=/usr/neos/PROBE/bin/pids/prog.pid
```

Sensors

```
alarm_service.path=./logs/sensor_state
alarm_service.sending_interval=60000
sensor.info_path=./logs/sensor_info
sensor.common.polling_interval=5000
sensor.common.preload_attempt_count=3
sensor.ram.averaging_interval=60000
sensor.fan_speed.min_allowed=600
sensor.fan_speed.max_allowed=18975
sensor.cpu_temp.min_allowed=0
sensor.cpu_temp.max_allowed=95
sensor.voltage.min_allowed=11.4
sensor.voltage.max_allowed=12.6
sensor.probes_state.max_allowed_net_speed=13107200
```

```
regional_distribution=true
```

3.2.2 Конфигурация пультов управления ОРМ (pu.config.properties)

Настройки, используемые при наличии у оператора связи узлов с территориально распределенной архитектурой, сохраняются в файле pu.config.properties.

Существует несколько режимов работы:

- Обычный.

При обычном режиме работы в конфигурационном файле должны быть перечислены ПУ. Формат записи:

```
pu.1
pu.2
pu.3
...
pu.N
```

Всем ПУ присваиваются номера от 1 до N, где N — номер последнего желаемого ПУ.

Примечание — Максимальное количество ПУ равно 100.

Для каждого ПУ будут настроены каналы управления и данных. Порты вычисляются по формуле:

$$PuPort = BasePort + (Puld - 1) * 32$$

Используемые параметры:

1. BasePort — это порты нулевого канала (для головного ПУ с номером 1). По умолчанию установлены значения: 16118 для канала управления и 16117 для канала данных.
2. Puld — это номер ПУ.

Параметры ПУ, относящиеся к режиму территориального распределения (ТР) не будут учитываться в обычном режиме.

- Территориальное распределение.

В данном режиме в конфигурационном файле должны быть перечислены ПУ аналогично тому, как это сделано в обычном режиме, но дополнительно будут добавлено два параметра:

1. Код региона, где расположен ПУ, перечисляются через запятую. Может быть указано несколько значений одновременно.
2. Настройка каналов дочерних ПУ для данного ПУ. Возможные значения: child_on (да) и child_off (нет).

Порты каналов дочерних ПУ вычисляются по формуле:

$$ChildPuPort = PuPort + VKSTid * 2$$

Используемые параметры:

1. PuPort — это порт головного ПУ.
2. VKSTid — порядковый номер дочернего пульта внутри группы (0 — для родительского, диапазон от 1 до 15 для дочерних).

Примечание — Для ПУ с номером 1 (головного) код региона не указывается, так как данный ПУ имеет доступ ко всем данным, как и его дочерние ПУ.

Пример файла конфигурации:

```
pu.1:child_off
pu.2:78:child_off
pu.3:77:child_off
pu.4:44:child_off
```

3.3 Настройка параметров приложения Voice-storage

Приложение Voice-storage предназначено для сбора и хранения голосовой информации и текстовых сообщений пользователей услуг.

Настройка приложения Voice-storage осуществляется в следующих конфигурационных файлах:

- trace.cfg — файл, содержащий настройки подсистемы записи логов и cdr-файлов;
- voice-storage.cfg — файл, содержащий основные настройки приложения.

3.3.1 Конфигурация параметров системы журналирования (trace.cfg)

В конфигурационном файле trace.cfg описываются параметры системы журналирования.

Конфигурация журналов может быть перезагружена средствами динамической команды: `./reload trace.cfg`

Примечание — Подробная информация о расшифровке файлов с отладочной информацией предназначена только для служебного пользования. Не рекомендуется менять значения данного файла, так как это может привести к некорректной работе приложения.

В таблице ниже описаны параметры конфигурационного файла.

Таблица 6 — Параметры trace.cfg

Параметр	Значимость параметра/ перезапуск	Описание
Секция [Trace]		
Common	O/R	Набор параметров. Общие настройки системы журналирования. Тип — object. Включает параметры: tracing, dir, no_signal.
tracing	M/R	Активность системы журналирования. Тип — bool. Возможные значения: 0 — система отключена (false); 1 — система активна (true). Значение по умолчанию: 1

Параметр	Значимость параметра/ перезапуск	Описание
dir	O/R	<p>Путь к каталогу, где будут храниться журналы. При необходимости система создаст недостающие каталоги.</p> <p>Тип — string.</p> <p>Если путь начинается с "./", то путь берётся относительно текущего каталога, если с "/", то от корня, иначе — от каталога по умолчанию. Путь может содержать ".." и маску формата времени.</p> <p>Значение по умолчанию: «./logs».</p>
no_signal	O/R	<p>Набор чисел через запятую или строка «all». Сигналы ОС Linux, не перехватываемые системой журналирования. Остальные сигналы система перехватывает и пишет об этом в основные журналы.</p> <p>Тип — list, элементы — int, разделитель — ",", запятая.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> "intercept all" — перехватывать все; "all" — вообще не перехватывать сигналы. <p>Значение по умолчанию: "intercept all" — перехватывать все.</p>
Параметры журналов		
logs	O/R	<p>Набор параметров. Конфигурация журналов.</p> <p>Состоит из параметров следующего формата: log_name = { список_параметров };</p> <p>где log_name — имя_журнала.</p> <p>Тип — object.</p> <p>Описание каждого журнала является опциональным.</p>

Параметр	Значимость параметра/ перезапуск	Описание
log_name	O/R	<p>Имя журнала. Допускаются символы, цифры, знак подчеркивания. Служит идентификатором журнала в системе.</p> <p>Тип — string.</p> <p>Существует несколько зарезервированных имен:</p> <p>trace — журнал «по умолчанию». Содержит параметры: file, mask, level, tee.</p> <p>info — журнал, отображающий информационные события: создание и течение времени жизни компонентов. Содержит параметры: file, mask, level.</p> <p>warning — журнал предупреждений. Содержит параметры: file, mask, level, tee.</p> <p>error — журнал, отображающий критические ошибки.</p> <p>out — журнал, отображающий данные по работе различных компонент приложения.</p> <p>RtpSender_trace — журнал, отображающий данные по отправке голосовых файлов на ПУ.</p> <p>RtpSender_warning — журнал предупреждений о некорректном преобразовании RTP потоков в звуковые дорожки.</p>
file	O/R	<p>Путь к файлу лога.</p> <p>Тип — string.</p> <p>При необходимости недостающие каталоги создаются. Допускается задание пустого имени файла, если level=0, в этом случае запись производится согласно параметру tee. В случае отсутствия этого параметра, запись на диск не производится.</p> <p>Если путь начинается с "./", то путь берётся относительно текущего каталога, если с "/", то от корня, иначе — от каталога по умолчанию. Путь может содержать ".." и маску формата времени.</p> <p>Пример: cdr/%Y/%m/%d/%H_%M_%S.log преобразуется в cdr/2004/07/07/13_54_31.log.</p>
mask	O/R	<p>Маска формата вывода автоматических полей в журнале.</p> <p>Тип — string.</p> <p>Детальное описание смотри в подпункте 3.3.1.2 «Модификаторы mask Ошибка! Источник ссылки не найден.».</p>

Параметр	Значимость параметра/ перезапуск	Описание
level	O/R	Уровень журнала. Тип — int. Специфично для журнала. Примечание — Сообщения с уровнем, большим, чем level, игнорируются.
type	O/R	Тип журнала и дополнительные настройки. Тип — string. Детальное описание смотри в подпункте 3.3.1.4 « Ошибка! Источник ссылки не найден.Ошибка! Источник ссылки не найден. ».
separator	O/R	Разделитель автоматических полей. Тип — string. Значение по умолчанию: значение параметра common. Примечание — весь вывод времени (date, time, tick) рассматривается как одно поле.
period	O/R	Период обновления файла лога. Тип — object. Формат: «interval» + «shift», где interval — промежуток времени между соседними обновлениями; shift — первоначальный сдвиг. Детальное описание смотри в подпункте 3.3.1.3 « Ошибка! Источник ссылки не найден. ». Примечание — Сдвиг не может быть больше длины периода, и в случае некорректного значения игнорируется. Пример: day+3hour — файл обновляется каждый день в 03:00:00.
buffering	O/R	Настройки буферизированной записи. Тип — object. Подпараметры: cluster_size, clusters_in_buffer, overflow_action. Детальное описание смотри в подпункте 3.3.1.1 «Модификаторы buffering».

Параметр	Значимость параметра/ перезапуск	Описание
tee	O/R	<p>Дублирование потока вывода.</p> <p>Тип — string.</p> <p>Возможные значения: stdout, stderr, trace, info, warning или имя любого другого лога.</p> <p>Если перед именем написать «минус», например — «trace», то при дублировании не пишется имя исходного лога.</p> <p>Пример: tee=stdout & stderr & trace & info & warning or any your log.</p>
limit	O/R	<p>Ограничение на максимальное количество строк в файле (записей).</p> <p>Тип — int.</p> <p>Как только достигнут предел строк, лог автоматически открывается заново. При этом не исследуется реальное количество строк в файле на данный момент. Если имя файла зависит от времени, то открывается новый файл, иначе файл обнуляется.</p>

Пример файла конфигурации:

```
[Trace]
common={tracing=1; dir=./logs;}

logs=
{
  RtpSender_trace = {
    file = rtp_sender.log;
    mask = date & time & tick;
    level = 10;
  };

  RtpSender_warning = {
    file = rtp_sender_warning.log;
    mask = date & time & tick;
    level = 10;
  };

  trace = {
    file=trace.log;
```

```

mask=date & time & tick & pid & file;
level=10;
};
error = {
file=error.log;
mask=date & time & tick & pid & file;
level=10;
};
info = {
file=info.log;
mask=date & time & tick;
level=10;
tee=trace;
};
warning = {
file=warning.log;
mask=date & time & tick & pid & file;
level=9;
tee=trace;
};

```

3.3.1.1 Модификаторы buffering

Ниже в таблице описаны модификаторы параметра buffering.

Таблица 7 — Модификаторы buffering

Параметр	Описание
cluster_size	Размер кластера, измеряется в Кбайтах. Тип — int. Значение по умолчанию: 128 Кб.
clusters_in_buffer	Длина буфера в кластерах. Тип — int. Значение по умолчанию: 0.

Параметр	Описание
overflow_action	<p>Действие при переполнении буфера.</p> <p>Тип — string.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> erase — удаление; dump — запись. <p>Значение по умолчанию: dump.</p>

3.3.1.2 Модификаторы mask

Ниже в таблице описаны модификаторы параметра mask.

Таблица 8 — Модификаторы mask

Параметр	Описание
date	<p>Дата создания.</p> <p>Формат записи — DD/MM/YY.</p> <p>Тип — datetime.</p>
time	<p>Время создания.</p> <p>Формат записи — hh:mm:ss.</p> <p>Тип — datetime.</p>
tick	<p>Миллисекунды.</p> <p>Формат записи: если задано time — “.mss”, три цифры, а если не задано time — “.mssmss”, шесть цифр.</p> <p>Тип — int.</p>
state	<p>Состояние системы.</p> <p>Тип — int или string.</p>
pid	<p>Идентификатор процесса.</p> <p>Формат записи — шесть цифр.</p> <p>Тип — int.</p>
tid	<p>Идентификатор потока.</p> <p>Формат записи — шесть цифр.</p> <p>Тип — int.</p>
level	<p>Уровень журнала для записи.</p> <p>Тип — int.</p>
file	<p>Файл и строка в файле с исходным кодом, откуда производится вывод.</p> <p>Тип — string.</p>

3.3.1.3 Модификаторы period

Ниже в таблице описаны модификаторы параметра period.

Таблица 9 — Модификаторы period

Параметр	Описание
count	Текущее время для имени файла. Количество стандартных периодов. Тип — int. Значение по умолчанию: 1.
type	Вид временного интервала. Тип — string. Возможные значения: sec, min, hour, day, week, month и year.

3.3.1.4 Модификаторы type

Ниже в таблице описаны модификаторы параметра type.

Таблица 10 — Модификаторы type

Параметр	Описание
name_now	Текущее время для имени файла.
name_period	Время для имени файла, начало периода.
truncate	Файл при открытии обнуляется.
append	Файл при открытии не обнуляется, а дописывается.
log	Состоит из truncate и name_now, при падении пишется информация о сигнале.
cdr	Состоит из append и name_now, при падении не пишется информация о сигнале.

Три пары взаимоисключающих значений: log или cdr, truncate или append, name_now или name_period.

Возможно переназначить параметр, заданный по умолчанию, в другом модификаторе.

Примеры: type = cdr & name_period — cdr с именем файла по началу периода; type = append — log без обнуления файлов.

3.3.2 Настройка основных параметров приложения (voice-storage.cfg)

Настройки управления конфигурацией приложения сохраняются в файле voice-storage.cfg.

В таблице ниже описаны параметры конфигурационного файла.

Таблица 11 — Параметры voice-storage.cfg

Параметр	Значимость параметра/ перезапуск	Описание
Секция [App]		

Параметр	Значимость параметра/ перезапуск	Описание
DpdkDriver		Тип драйвера. Возможные значения: af и dpdk. Тип — string.
RtpInterface		Сетевой интерфейс, осуществляющий прием RTP трафика. Тип — string.
PcapRoot		Директория, в которой хранятся промежуточные файлы. Тип — string.
ReceiverCoreNum		type: uint8_t
BufferSize		type: uint32_t;
ChunkSize		
AccessPointPort		Порт для Thrift-интерфейса. Тип — int.
MuhtarPort		
NumberOfWorkers		Число потоков. Тип — int.
RtpWaitTime		
AsnContextTimeout		
FlushPaths		Директория, в которой хранятся звуковые дорожки. Тип — string.
ReceiverCoreNum		Номер ядра, которое будет использоваться для обработки трафика. Тип — int.

Пример файла конфигурации:

```
[App]
DpdkDriver="af";
RtpInterface={"rtp_in"};
PcapRoot="/usr/neos/data/dumps/";
ReceiverCoreNum=1;
BufferSize=80;
ChunkSize=10;
```

```
AccessPointPort=12345;
MuhtarPort=12346;
NumberOfWorkers=1;
RtpWaitTime=600;
AsnContextTimeout=1800;
FlushPaths = {
  "/usr/neos/data/flushed/";
};
```

3.4 Настройка параметров приложения VOICE-PROBE

Приложение VOICE-PROBE предназначено для перехвата трафика и обработки перехваченных сообщений.

Данное приложение производит фильтрацию трафика, извлекая сигнальную информацию вызова (DSS1, ISUP, SIP), с определенных точек сети. Отфильтрованная сигнальная информация перенаправляется внутренним обработчиком. Обработчики осуществляют разбор сообщений сигнализации.

Настройка приложения VOICE-PROBE осуществляется в следующих конфигурационных файлах:

- alarm/ — настройка параметров подсистемы обработки аварий. Содержит следующие файлы: ap.cfg и ap_dictionary;
- app.cfg — файл, содержащий основные настройки приложения;
- component/ — настройка аппаратного обеспечения для приема потоков E1. Содержит следующие файлы: config.cfg и physical.cfg;
- monitor.cfg — файл, содержащий настройки мониторинга физического уровня;
- om_interface.cfg — файл, содержащий настройки подключений протокола OMI;
- tdm.cfg — файл, содержащий настройки параметров обработки данных в потоках E1;
- Threads.cfg — файл, содержащий настройки многопоточности;
- trace.cfg — файл, содержащий настройки подсистемы записи логов и cdr-файлов;
- VOP.cfg — файл, содержащий настройки параметров RTP-сессий.

3.4.1 Настройка подсистемы Alarm Processor

Для настройки подсистемы Alarm Processor существуют два файла:

- ap.cfg — файл конфигурации подсистемы Alarm Processor;
- ap_dictionary — словарь подсистемы Alarm Processor.

Файлы ap.cfg и ap_dictionary находятся в директории /usr/neos/PROBE/config/alarm/.

Файл ap.cfg содержит параметры подсистемы Alarm Processor, параметры SNMP-соединения и правила преобразования компонентных адресов в SNMP-адреса.

В файле ap_dictionary находятся соответствия между значениями переменных и идентификаторами трапов. Идентификаторы трапов используются SNMP-менеджером для соответствующей их обработки.

Файл конфигурации ap.cfg

В таблице ниже приведено описание секций, из которых состоит файл ap.cfg.

Примечание — Не рекомендуется менять параметры в данном конфигурационном файле.

Таблица 12 — Состав секций файла конфигурации ar.cfg

Параметр	Значимость параметра/ перезапуск	Описание
Секция [General]. Общие параметры обработки аварий. Параметры из данной секции не требуют какого-либо редактирования.		
Root	O/R	Корень каталога. Тип — string.
ApplicationAddress	M/R	Адрес приложения. Тип — string.
MaxConnectionCount	O/R	Максимальное количество одновременных подключений. Тип — int. Значение по умолчанию: 10
ManagerThread	O/R	Запуск подсистемы Alarm Processor в отдельном потоке приложения. Тип — bool. Возможные значения: 0 — работа в основном потоке; 1 — работа в отдельном потоке. Значение по умолчанию: 0 Примечание — если этот параметр опущен, интегрированный менеджер не будет создан. Значение параметра «ManagerThread» зависит от загрузки процессора. Если загрузка процессора достаточно большая, то подсистему Alarm Processor рекомендуется запускать в отдельном потоке (ManagerThread=1).
CyclicWalkTree	O/R	Циклический обход древа. Тип — bool. Возможные значения: 0 — нет; 1 — да. Значение по умолчанию: 0

Параметр	Значимость параметра/ перезапуск	Описание
UseATE_MultipleIndexation	O/R	Формирование индекса из чисел, содержащихся в СА. Тип — bool. Возможные значения: 0 — не формировать (индекс автоинкрементный); 1 — формировать. Значение по умолчанию: 0
Секция [Dynamic]. Набор параметров. Список переменных и их величин, которые указывают, что динамические объекты должны быть удалены. Формат: {variable_address;value;}		
variable_address	O/R	Компонентный адрес переменной Тип — string.
value	O/R	Значение переменной. Тип — string.
Секция [SNMP]. Сетевые параметры протокола SNMP.		
ListenIP	O/R	IP-адрес, с которым будет устанавливаться соединение система обработки сообщений (AlarmProcessor). Тип — string. Значение по умолчанию: 0.0.0.0
ListenPort	O/R	Номер порта, с которым будет устанавливаться соединение система обработки сообщений (AlarmProcessor). Тип — int. Диапазон значений: 0 — 65535 Значение по умолчанию: 161
OwnEnterprise	O/R	SNMP-адрес приложения. Тип — string.
Секция [StandardMib] Определяет список стандартных переменных и их значений. Формат записи, описывающей стандартную переменную: {#addrSNMP;#typeVar;#value;}		
addrSNMP	O/R	Адрес SNMP для переменной. Тип — string.

Параметр	Значимость параметра/ перезапуск	Описание
typeVar	O/R	Тип переменной. Тип — string.
value	O/R	Значение переменной. Тип — string.
<p>Секция [AtePath2ObjName]</p> <p>Правила преобразования компонентного адреса переменной в SNMP-адрес.</p> <p>Формат правила: {#ctObject; #caVar;}</p> <p>Для каждого типа объекта необходимо прописать компонентный адрес, иначе объекты не будут добавляться в SNMP-дерево.</p>		
ctObject	O/R	Компонентный тип объекта. Тип — regex.
caVar	O/R	Компонентный адрес переменной. Тип — string.
<p>Секция [AtePath2Oid]</p> <p>Набор параметров. Описывает правила замены ATE-пути в SNMP-пути из стандартного мибя.</p> <p>Формат правила: {#ctObject;#caVar;#ObjectID;}. Для каждого типа объекта необходимо прописать адрес CA(1), иначе объекты не будут добавляться в SNMP-дерево.</p>		
ctObject	O/R	Компонентный тип объекта. Тип — regex.
caVar	O/R	Компонентный адрес переменной. Тип — string.
ObjectID	O/R	SNMP-идентификатор. Тип — int.
<p>Секция [SNMPTrap]</p> <p>Правила отправки SNMP-сообщений (далее по тексту — трапов).</p> <p>Определяются параметры взаимодействия с SNMP-менеджерами. Подсистема Alarm Processor может взаимодействовать с несколькими SNMP-менеджерами одновременно. Для каждого SNMP-менеджера в секции [SNMPTrap] можно определить свои параметры.</p> <p>Формат записи: { #ipManSNMP;#portManSNMP;#caObjFilter;#ctObjFilter;#caVarFilter; }</p>		
ipManSNMP	O/R	IP-адрес SNMP-менеджера. Тип — string.

Параметр	Значимость параметра/ перезапуск	Описание
portManSNMP	O/R	Порт SNMP-менеджера. Тип — int. Диапазон значений: 0 — 65535.
caObjFilter	O/R	Фильтр по адресу объекта. Тип — regex.
ctObjFilter	O/R	Фильтр по типу объекта. Тип — regex.
caVarFilter	O/R	Фильтр по адресу переменной. Тип — regex.
<p>Секция [Filter]</p> <p>Правила фильтрации Alarm-сообщений.</p> <p>Определяются фильтры по индикатору трапа и по индикатору динамического объекта. Эти фильтры «отсеивают» трапы на входе подсистемы Alarm Processor, то есть между логикой, которая является источником трапов и подсистемой Alarm Processor.</p>		
CA_Object	O/R	Фильтр по адресу объекта. Тип — regex. Значение по умолчанию: ".*"
CT_Object	O/R	Фильтр по типу объекта. Тип — regex. Значение по умолчанию: ".*"
CA_Var	O/R	Фильтр по адресу переменной. Тип — regex. Значение по умолчанию: ".*"
TrapIndicator	O/R	Фильтр по индикатору трапа. Тип — string. Значение по умолчанию: 1
DynamicIndicator	O/R	Фильтр по динамическому индикатору объекта. Тип — string. Значение по умолчанию: 0
<p>Секция [SpecificTrapCA_Object]</p> <p>Набор параметров. Правила создания идентификаторов SNMP трапов. Описывается соответствие идентификатора трапа адресу компонента (CA).</p> <p>Формат: {#caVar;#specificTrapOffset;}.</p>		

Параметр	Значимость параметра/ перезапуск	Описание
caVar	O/R	Компонентный адрес переменной. Тип — string.
specificTrapOffset	O/R	Смещение в нумерации. Тип — int.
<p>Секция [SpecificTrapCT_Object]</p> <p>Набор параметров. Правила создания идентификаторов SNMP трапов. Описывается соответствие идентификатора трапа типу компонента (CT).</p> <p>Формат: { #ctObject;#specificTrapBase; }</p>		
ctObject	O/R	Компонентный тип объекта. Тип — regex.
specificTrapBase	O/R	Число начала нумерации. Тип — int
<p>Секция [SpecificTrapCA_Var]</p> <p>Набор параметров. Правила создания идентификаторов SNMP трапов. Описывается соответствие идентификатора трапа компонентному адресу переменной.</p> <p>Формат: { #caObj;#specificTrapOffset; }</p>		
caObj	O/R	Компонентный адрес объекта. Тип — regex.
specificTrapOffset	O/R	Смещение в нумерации. Тип — int.
<p>Секция [Logs]</p>		
TreeTimerPeriod	O/R	Период хранения текущего состояния всех объектов в журнале (в миллисекундах). Тип — int. Значение по умолчанию: 60000 мс.
FilterLevel	O/R	Правила аварийной фильтрации журналов (логов). Тип — list, элементы — строки типа object. Формат записи: { #caObj;#ctObj;#caVar;#nLevel; }
caObj	O/R	Компонентный адрес объекта. Тип — regex.

Параметр	Значимость параметра/ перезапуск	Описание
ctObj	O/R	Компонентный тип объекта. Тип — regex.
caVar	O/R	Компонентный адрес переменной. Тип — regex.
nLevel	O/R	Уровень журнала. Тип — int.

Пример файла конфигурации:

```
[General]
ApplicationAddress = "DeMUX.1";
MaxConnectionCount = "100";
ManagerThread = "0";
CyclicTreeWalk = "0";
[Dynamic]
[AtePath2ObjName]
{
"Ph(100,1).Card(1,1)";
"CA(4096)";
};
{
"Ph(100,1).Card(1,1)";
"OSTATE(4097)";
};
{
"Ph(100,1).Card(1,1)";
"ASTATE(4098)";
};
{
"Ph(100,1).Trunk(2,1)";
"CA(4096)";
};
{
"Ph(100,1).Trunk(2,1)";
"OSTATE(4097)";
};
```

```
};  
[SNMP]  
ListenIP = "0.0.0.0";  
ListenPort = "1161";  
OwnEnterprise = "1.3.6.1.4.1.20873.100";  
[StandardMib]  
{  
  "1.3.6.1.2.1.1.1.0";  
  "STRING";  
  "MAK";  
};  
{  
  "1.3.6.1.2.1.1.2.0";  
  "OBJECT_ID";  
  "1.3.6.1.4.1.20873.100";  
};  
[AtePath2Oid]  
[SNMPTrap]  
PrefixCA_Trap = "ITG";  
{  
  "127.0.0.1";  
  "162";  
  ".*";  
  ".*";  
};  
[Filter]  
CA_Object = ".*";  
CT_Object = ".*";  
CA_Var = ".*";  
TrapIndicator = "-1";  
DynamicIndicator = "-1";  
[SpecificTrapCA_Object]  
{  
  "Ph.Card.O$";  
  "1";  
};  
[SpecificTrapCT_Object]  
{
```

```
"Ph.Card.Alter";
"2";
};
[SpecificTrapCA_Var]
{
"Warn.Config.Invalid";
"100";
};
{
"Alarm.Route";
"101";
};
[Logs]
TreeTimerPeriod = "0";
FilterLevel={
{
".*";
"^Ph.*";
"OSTATE";
"1";
};
{
".*";
"^Ph.*";
"STATE";
"1";
};
};
```

Правила назначения трапов SpecificTraps (ap_dictionary)

В словаре ap_dictionary указываются значения SpecificTrap для зарезервированных переменных OSTATE, ASTATE и HSTATE.

Пример файла конфигурации:

```
OSTATE = {
  {1; ACTIVATE}; SP_Trap = 1;
  {0; FAIL}; SP_Trap = 2;
};
ASTATE =
{
  {1; UNBLOCKED}; SP_Trap=3;
  {2; BLOCKED}; SP_Trap=4;
};
HSTATE =
{
  {1; ON}; SP_Trap=5;
  {0; OFF}; SP_Trap=6;
};
```

Алгоритм формирования идентификаторов трапа

В словаре ap_dictionary указываются значения SpecificTrap для зарезервированных переменных OSTATE, ASTATE и HSTATE.

В формировании идентификатора трапа участвует информация, содержащаяся в файлах ap.cfg и ap_dictionary.

При возникновении какого-либо события подсистема Alarm Processor получает адрес компонента источника события и переменную, связанную с событием. Далее, используя информацию файлов ap.cfg и ap_dictionary, подсистема Alarm Processor вычисляет значение идентификатора трапа.

Секции [SpecificTrapCA_Object] и [SpecificTrapCT_Object] файла ap.cfg предоставляют информацию для вычисления базового значения идентификатора трапа, файл ap_dictionary и секция [SpecificTrapCA_Var] файла ap.cfg — смещения.

В условной форме формулу вычисления идентификатора трапа можно записать: $\text{trap_id} = \text{база} * 1000 + \text{смещение}$.

Таким образом, алгоритм формирования идентификатора трапа следующий:

- **базовое значение идентификатора трапа** ищем по адресу компонента источника события в секции [SpecificTrapCA_Object], если не нашли — по типу компонента источника события в секции [SpecificTrapCT_Object] (по умолчанию, «база» — 0);
- **смещение идентификатора трапа** ищем по значению переменной в ap_dictionary, если не нашли — по названию в [SpecificTrapCA_Var], если и там нет — 0;
- **идентификатор трапа** складывается из базового значения, умноженного на 1000, и смещения.

Рассмотрим работу алгоритма на конкретном примере. Допустим, в файле ap.cfg представлено следующее содержимое секций [SpecificTrapCA_Object], [SpecificTrapCT_Object] и [SpecificTrapCA_Var]:

```
[SpecificTrapCA_Object]
{"Ph.Card.0$"; 1};

[SpecificTrapCT_Object]
{"Ph.Card.Alter"; 2};
{"Ph.Card.ADSP"; 3};

[SpecificTrapCA_Var]
{"Warn.Config.Invalid"; 100};
{"Alarm.Route"; 101};

Файл ar_dictionary содержит следующие данные:

OSTATE =
{
    1; SP_Trap = 1;
    0; SP_Trap = 2;
    -1; SP_Trap = 3;
};

ASTATE =
{
    1; SP_Trap = 4;
    0; SP_Trap = 5;
    -1; SP_Trap = 6;
};
```

Предположим, что в подсистему Alarm Processor от логики поступило событие об изменении оперативного состояния (переменная OSTATE) компоненты с адресом «Ph.Card.0». Подсистема Alarm Processor начинает поиск в файле ar.cfg, в секции [SpecificTrapCA_Object]. В ней обнаруживается запись («{"Ph.Card.0\$"; 1};») с искомым адресом. Эта запись содержит соответствующий идентификатор — «1». На этом поиск в файле ar.cfg прекращается. Найденный идентификатор «1» умножается на «1000» в результате получаем базовое значение идентификатора — «1000». Далее подсистема Alarm Processor выполняет поиск переменной с именем «OSTATE» в файле ar_dictionary. Поиск завершается с положительным результатом — блок «OSTATE». Блок «OSTATE» в примере выше содержит три записи, каждая из которых состоит из значения переменной и соответствующего идентификатора. Результатом поиска будет идентификатор, соответствующий значению переменной «OSTATE». Допустим переменная «OSTATE» имеет значение «-1», ему в примере соответствует идентификатор «3».

В завершение складываем базовое значение идентификатора («1000») с результатом поиска в файле ar_dictionary — «3», получаем «1003». Это и будет итоговое значение идентификатора трапа.

Примечание — Отрицательному результату поиска в файле ar_dictionary соответствует значение «0», то есть итоговое значение идентификатора трапа будет совпадать с базовым значением, если взять пример выше — «trap_id = 1000 + 0».

3.4.2 Настройка основных параметров приложения (app.cfg)

Настройки управления конфигурацией приложения сохраняются в файле app.cfg.

В таблице ниже описаны параметры конфигурационного файла.

Таблица 13 — Параметры app.cfg

Параметр	Значимость параметра/ перезапуск	Описание
Секция [PacketEngine]		
inputDpdkDriver	M/R	Тип интерфейса для пассивного приема данных. Тип — string. Возможные значения: af — работа с сетью на уровне интерфейсов; dpdk — работа с сетью на уровне устройств; pcap — работа с pcap-файлами.
outputDpdkDriver	M/R	Тип интерфейса для отправки данных на RTP-storage. Тип — string. Возможные значения: af — работа с сетью на уровне интерфейсов; dpdk — работа с сетью на уровне устройств; pcap — работа с pcap-файлами.
inputInterface	M/R	Интерфейс для пассивного приема данных. Тип — string. Возможные значения: af — имя сетевого интерфейса; dpdk — PCI адрес интерфейса с драйвером dpdk; pcap — имя pcap-файла.
outputInterfac	M/R	Интерфейс для отправки данных на RTP-storage. Тип — string. Возможные значения: af — имя сетевого интерфейса; dpdk — PCI адрес интерфейса с драйвером dpdk; pcap — имя pcap-файла
receiverCoreNum	O/R	Номер ядра CPU, который осуществляет прием данных от точек консолидации. Тип — int.
senderCoreNum	O/R	Номер ядра CPU, который осуществляет отправку данных на RTP-storage. Тип — int.

Параметр	Значимость параметра/ перезапуск	Описание
workerCoreNum	O/R	Номер ядра CPU, к которому привязывается поток обработчик пакетов. Тип — int.
workerRxQueueSize	O/R	Размер очереди пакетов, поступающих к обработчику от приемника. Тип — int.
Секция [General]		
enable645	M/R	Включить функцию поддержки 645 приказа Минкомсвязи России. Тип — bool. Возможные значения: 1 — включить; 0 — не включать. Значение по умолчанию: 0
enableDSS1	M/R	Обработка DSS1 сигнализации. Тип — bool. Возможные значения: 1 — включить; 0 — не включать. Значение по умолчанию: 1
enableISUP	M/R	Обработка ISUP сигнализации. Тип — bool. Возможные значения: 1 — включить; 0 — не включать. Значение по умолчанию: 0
enableH323	M/R	Обработка H323 сигнализации. Тип — bool. Возможные значения: 1 — включить; 0 — не включать. Значение по умолчанию: 0

Параметр	Значимость параметра/ перезапуск	Описание
enableSIP	M/R	Обработка SIP сигнализации. Тип — bool. Возможные значения: 1 — включить; 0 — не включать. Значение по умолчанию: 0
Секция [DSS1-Probe] Настройка параметров обработки DSS1 сигнализации		
ENABLED	M/R	Обработка DSS1 сигнализации. Тип — bool. Возможные значения: 1 — включить; 0 — не включать. Значение по умолчанию: 1
maxCallDurationSec	M/R	Максимальная длительность вызова (в секундах). Тип — int. При достижении указанного значения вызов завершается принудительно.
tdmSettingsFile	M/R	Путь к файлу конфигурации с настройками E1. Тип — string.
Секция [ISUP-Probe] Настройка параметров обработки ISUP сигнализации. Включает в себя параметры: «ENABLED», «maxCallDurationSec», «tdmSettingsFile». Описание аналогично приведённому выше для секции [DSS1-Probe].		
Секция [VoiceManager]		
destinationIP	M/R	IP-адрес шлюза, который получает RTP-пакеты от компоненты обработки трафика. Тип — string.
destinationPort	M/R	Порт, на который будут приходить данные от компоненты обработки трафика. Тип — int.

Параметр	Значимость параметра/ перезапуск	Описание
Секция [FinSender] секция настройки отправки специального пакета, который voice-probe должен отправлять на RTP-Storage по завершению вызова. Данный пакет указывает на то, что контент по вызову пора переносить из временного хранилища в постоянное.		
times	O/R	Количество попыток повторной отправки пакета. Тип — int.
interval	O/R	Временной интервал до повторной отправки пакета (в миллисекундах). Тип — int.
Секция [CDR-Writer] Параметры подключения к БД Postgres для хранения записей (CDR) по вызовам.		
Host	O/R	Имя хоста (или IP-адрес) БД, в которой будут храниться записи по вызовам. Тип — string.
User	O/R	Имя пользователя для подключения к БД. Тип — string.
Database	O/R	Имя БД, к которой осуществляется подключение. Тип — string.
Password	O/R	Пароль для доступа к БД. Тип — string.
Table	O/R	Раздел таблицы, в котором отображаются записи с вызовами. Тип — string.

Пример файла конфигурации:

<pre>[PacketEngine] inputDpdkDriver = af; outputDpdkDriver = af; inputInterface = enp2s0.3; outputInterface = grelan; [General] enable645 = 0; enableDSS1 = 1;</pre>

```

enableISUP = 0;
enableH323 = 0;
enableSIP = 0;

[DSS1-Probe]
ENABLED      = 1;
maxCallDurationSec = 60;
tdmSettingsFile = "./config/tdm.cfg";

[ISUP-Probe]
ENABLED      = 0;
maxCallDurationSec = 60;
tdmSettingsFile = "./config/tdm.cfg";

[VoiceManager]
destinationIP = 192.0.2.1;
destinationPort = 9;

[FinSender]
times=3;
interval=500;

[CDR-Writer]
Host = "172.16.202.103"
User = "sorm";
Database = "jarovaya_ats"
Password = "elephant";
Table = "records";

```

3.4.3 Настройка мониторинга физического уровня (monitor.cfg)

Параметры Monitor DI настраиваются в файле конфигурации config/monitor.cfg.

В таблице ниже описаны параметры данного конфигурационного файла.

Таблица 14 — Параметры monitor.cfg.

Параметр	Значимость параметра/ перезапуск	Описание

Параметр	Значимость параметра/ перезапуск	Описание
Enable	O/P	<p>Включение/выключение монитора. Может быть установлен как для каждого фильтра персонально, так и глобально для всех фильтров.</p> <p>Тип — bool.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> 0 — выключен; 1 — включен. <p>Фильтр будет считаться активным, если глобальный Enable и его локальный Enable будут выставлены в 1. В противном случае фильтр исключается из конфигурации.</p>
Duration	O/P	<p>Количество минут, в течении которых будет действовать данная конфигурация с момента ее применения.</p> <p>Тип — int.</p> <p>Значение по умолчанию: 0 (бесконечно)</p>
BM_BufSize	O/P	<p>Включение/выключение буферизации в бинарном мониторе.</p> <p>Тип — int.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> 0 — выключена; 100 – 10000 — включена, при этом значение параметра определяет размер буфера в байтах. <p>Значение по умолчанию: 100</p>
Filters	M/P	<p>Набор фильтров компонент, определяющих, какие из них должны быть записаны в monitor.log.</p>
<p>Секция [Base]</p> <p>Выполняет функцию фильтра тех компонент, которые будут мониториться. Состоит из «Ph.Card.[0-9]» и группы параметров, определяющих предмет мониторинга.</p> <p>«Ph.Card.[0-9]» — это база, относительно которой перечисляются типы устройств (предок для группы компонент).</p>		
Enable	M/P	<p>Локальный флаг для включения/выключения монитора.</p> <p>Тип — bool.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> 0 — выключен; 1 — включен. <p>Устанавливается для каждого фильтра персонально.</p>
Type	M/P	<p>Тип компонента, который будет мониториться.</p> <p>Тип — string.</p>

Параметр	Значимость параметра/ перезапуск	Описание
Name	M/P	Тип информации, которая будет попадать в лист мониторинга. Тип — string.
Path	M/P	Путь к тракту. Тип — string.

Пример файла конфигурации:

```
{
  Enable = "1";
  Duration = "0";
  BM_BufSize = "10000";
  Filters={
    Base = "Ph.Card.[0-9]+";
    {
      Enable = "1";
      Type = "Trunk";
      Name = "trunk";
      Path={
        "Trunk.[0-9]+";
      };
    };
    {
      Enable = "0";
      Type = "ITC";
      Name = "cards";
      Path={
        "ITC.[0-9]+";
      };
    };
    {
      Enable = "0";
      Type = "ITC";
      Name = "voip_dsp";
      Path={
        "ITC.[0-9]+.SHARC.[0-9]+";
      };
    };
  };
};
```

};

3.4.4 Настройка параметров взаимодействия (om_interface.cfg)

Настройки параметров соединения между функциональными модулями сохраняются в файле конфигурации с именем om_interface.cfg.

В таблице ниже описаны параметры данного конфигурационного файла.

Таблица 15 — Параметры om_interface.cfg

Параметр	Значимость параметра/ перезапуск	Описание
Секция [Dynamic]		
IP	O/R	IP-адрес динамического OM-сервера. Тип — string.
Port	O/R	Номер порта динамического OM-сервера. Тип — int. Значение по умолчанию: 0
Timers	Набор параметров. Описывает временные интервалы.	
SessionResponseTimeOut	O/R	Максимальное время существования сессии, по истечении которого сессия автоматически закрывается (в миллисекундах). Тип — int. Значение по умолчанию: 60000
TransactionResponseTimeOut	O/R	Максимальное время существования транзакции, по истечении которого транзакция автоматически закрывается (в миллисекундах). Тип — int. Значение по умолчанию: 30000
SegmentResponseTimeOut	O/R	Временной интервал ожидания ответа на отправленный запрос (в миллисекундах). Тип — int. Значение по умолчанию: 4000
MaxSegmentErrorCount	M/R	Количество ошибок (на отправленный запрос не пришел ответ в течение интервала SegmentResponseTimeOut), при достижении которого сетевая логика закрывает текущую сессию. Тип — int. Значение по умолчанию: 3

Параметр	Значимость параметра/ перезапуск	Описание
LoginReqTimeOut	M/R	<p>Временной интервал после установления соединения (в миллисекундах).</p> <p>Тип — int.</p> <p>В случае отсутствия сообщения LoginReq по истечении временного интервала LoginReqTimeOut сетевая логика разорвет tcp-соединение.</p> <p>Значение по умолчанию: 10000</p>
ReconnectTimeOut	M/R	<p>Временной интервал, по истечении которого клиентская сетевая логика возобновит попытку соединения (в миллисекундах).</p> <p>Тип — int.</p> <p>Значение по умолчанию: 1000</p>
KeepAliveTimeOut	M/R	<p>Временной интервал в рамках tcp-соединения, по истечении которого в случае сетевой активности будет отправлено сообщение KeepAliveReq (в миллисекундах).</p> <p>Тип — int.</p> <p>Значение по умолчанию: 30000</p>
KeepAliveResponseTimeOut	M/R	<p>Временной интервал ожидания KeepAliveAck на KeepAlive по истечению которого tcp-соединение будет автоматически разорвано (в миллисекундах).</p> <p>Тип — int.</p> <p>Значение по умолчанию: 10000</p>
Секция [Automatic]		
Timers	Набор параметров. Описание аналогично набору параметров «Timers» в секции [Dynamic].	
TransactionResponseTimeOut	O/R	<p>Максимальное время существования транзакции, по истечению которого транзакция автоматически закрывается (в миллисекундах).</p> <p>Тип — int.</p> <p>Значение по умолчанию 0</p>
SegmentResponseTimeOut	O/R	<p>Временной интервал ожидания ответа на отправленный запрос (в миллисекундах).</p> <p>Тип — int.</p> <p>Значение по умолчанию 0</p>

Параметр	Значимость параметра/ перезапуск	Описание
ReconnectTimeOut	M/R	Временной интервал, по истечении которого клиентская сетевая логика возобновит попытку соединения (в миллисекундах). Тип — int.
KeepAliveTimeOut	M/R	Временной интервал в рамках tcp-соединения, по истечении которого в случае сетевой активности будет отправлено сообщение KeepAliveReq (в миллисекундах). Тип — int. Значение по умолчанию: 60000
KeepAliveResponseTimeOut	M/R	Временной интервал ожидания KeepAliveAck на KeepAlive, по истечению которого tcp-соединение будет автоматически разорвано. Тип — int. Значение по умолчанию: 30000
Sockets	Набор параметров автоматического соединения с сервером. Включает следующие параметры: "Address", "Port".	
Address	O/P	IP-адрес, разрешённый клиенту (где 0.0.0.0 — любой). Тип — string.
Port	O/P	Номер порта. Тип — int.
ConnectionLogics	Набор параметров. Список сетевых логик. Описание сетевой логики, включает следующие параметры: "Priority", "MaxTransactionCount", "Login", "Password".	
CL.<number>	Набор параметров. Номер сетевой логики.	
Priority	M/P	Приоритет сетевой логики при распределении новой транзакции. Тип — int.
MaxTransactionCount	M/P	Максимальное количество транзакций, одновременно обрабатываемых данной сетевой логикой. Тип — int.
IP	M/P	IP-адрес сервера автосоединения. Тип — string.

Параметр	Значимость параметра/ перезапуск	Описание
Port	M/P	Номер порта сервера автосоединения. Тип — int.
Login	O/P	Логин логики. Тип — string.
Password	O/P	Пароль логики. Тип — string.
Directions		Список направлений автосоединения. Описание направления включает в себя следующие параметры: "CL_Monitoring", "ChangeOverTimeOut", "Primary".
Dir.<>		Строка. Название направления.
CL_Monitoring	O/P	Наблюдение за сетевыми логиками. Тип — bool. Возможные значения: 0 — сетевая логика не посылает данные примитивы; 1 — сетевая логика будет посылать примитив OM_CONNECTION_RESUME_IND в случае установления соединения и OM_CONNECTION_PAUSE_IND в случае разрыва соединения. Значение по умолчанию: 0
ChangeOverTimeOut	O/P	Временной интервал ожидания восстановления первичных соединений после их отключения, по истечении которого передача входящих запросов будет осуществляться на вторичные соединения. Тип — int. Значение по умолчанию: 0
Primary		Набор параметров первичного соединения. Включает в себя параметры: «MaxTraffic», «Connections».
MaxTraffic	M/P	Максимальное количество транзакций, одновременно обрабатываемых на данном направлении. Тип — int.
Connections	M/P	Перечень сетевых логик. Формат параметра: {"Number1";... "NumberN";}, где «NumberX» — номер сетевой логики.

Параметр	Значимость параметра/ перезапуск	Описание
Secondary		Оptionальный перечень параметров резервного соединения (Reloadable). Описание аналогично приведённому выше для «Primary».

Пример файла конфигурации:

```
[Dynamic]
Port = 15000;

Timers=
{
LoginReqTimeOut      = 10000;
ReconnectTimeOut     = 1000;
KeepAliveTimeOut     = 30000;
KeepAliveResponseTimeOut = 10000;
}

[Automatic]

Timers=
{
TransactionResponseTimeOut = 3600000;
SegmentResponseTimeOut     = 60000;
LoginReqTimeOut            = 10000;
KeepAliveTimeOut           = 30000;
KeepAliveResponseTimeOut   = 10000;
}

ConnectionLogics=
{
CCL.0=
{
MaxTransactionCount=5000;
IP_Address=192.168.90.27;   #SORM Server IP
Port=14567;
Login=Login;
Password=Password;
```

```
ServiceID=VOP_MI0;
}

SCL.0 =
{
Priority = 1; #SCL - server connection logic
MaxTransactionCount = 1000000;
IP    = 0.0.0.0;
Port  = 17531;
UserList = { {Login = "Login";Password = "Password";}};
}
}
Directions=
{

Dir.VOP_MI.0 =
{
Primary=
{
Connections={CCL.0; };
}
}
}
Services =
{
Service.VOP_MI0 =
{
MaxTraffic = 1000000;
Connections = { SCL.0 };
}
}
```

3.4.5 Настройка параметров для приема данных потока E1 (tdm.cfg)

Настройки параметров приема данных передаваемых в потоках E1 сохраняются в файле конфигурации с именем tdm.cfg.

В таблице ниже описаны параметры данного конфигурационного файла.

Таблица 16 — Параметры tdm.cfg.

Параметр	Значимость параметра/ перезапуск	Описание
Секция [DSS1] Настройка параметров для приема данных передаваемых по протоколу сигнализации DSS1.		
HDLC	Набор параметров. Настройка подуровня HDLC. Тип — object.	
ComponentAddr	M/R	Адрес компоненты HDLC. Тип — string.
Params — составной параметр, содержит подпараметр «Screening».		
Screening	O/R	Отсеивание информации при выводе в лог-файл. Тип — bool. Возможные значения: 0 — не производится; 1 — отсеивание производится. Значение по умолчанию: 1
Links	Набор параметров. Настройка потоков, по которым передается сигнальная информация. Тип — object.	
ID	O/R	Идентификатор пары потоков с сигнальной информацией. Тип — int.
Links	O/R	Компонентный адрес. Список компонент, с которых поступает сигнальная информация на Probe. Первый адрес компоненты в паре будет иметь направление в сторону станции, а второй — от станции. Тип — string.
Секция [ISUP] Настройка параметров для приема данных передаваемых по протоколу сигнализации ISUP		
HDLC	Набор параметров. Настройка подуровня HDLC. Тип — object. Описание аналогично набору параметров в секции [DSS1].	
Linksets	Набор параметров. Настройка потоков, по которым передается сигнальная информация. Тип — object.	

Параметр	Значимость параметра/ перезапуск	Описание
ID	O/R	Идентификатор пары потоков с сигнальной информацией. Тип — int.
Links	O/R	Компонентный адрес. Список компонент, с которых поступает сигнальная информация на Probe. Первый адрес компоненты в паре будет иметь направление в сторону станции, а второй — от станции. Тип — string.
CICs		Подсекция, описывает соответствие кода-идентификатора канала (CIC) протокола ISUP и таймслотов, в которых поступает сигнальная информация на Probe Тип — object.
ID	O/R	Идентификатор пары потоков с сигнальной информацией. Тип — int.
Links	O/R	Компонентный адрес. Список компонент, в которых передается сигнальная информация на Probe. Тип — string.

Пример файла конфигурации:

```
[DSS1]
HDLC = {
  Ph.Card.0.Trunk.4.TSL.16 = { params = "Screening=1;" };
  Ph.Card.0.Trunk.5.TSL.16 = { params = "Screening=1;" };
}

Links = {
  { ID = 1; Ph.Card.0.Trunk.4; Ph.Card.0.Trunk.5; };
}

[ISUP]
HDLC = {
  Ph.Card.0.Trunk.0.TSL.1 = { params = "Screening=1;" };
  Ph.Card.0.Trunk.1.TSL.1 = { params = "Screening=1;" };
}

Linksets = {
  {
    ID = 1;
  }
}
```

```

Links = {
  { ID = 1; Ph.Card.0.Trunk.0; Ph.Card.0.Trunk.1; };
}
CICs = {
  { ID = 2; Ph.Card.0.Trunk.0.TSL.2; Ph.Card.0.Trunk.1.TSL.2; };
  { ID = 3; Ph.Card.0.Trunk.0.TSL.3; Ph.Card.0.Trunk.1.TSL.3; };
  { ID = 4; Ph.Card.0.Trunk.0.TSL.4; Ph.Card.0.Trunk.1.TSL.4; };
}
}

```

3.4.6 Настройка многопоточности (Threads.cfg)

Настройки параметров для многопоточности сохраняются в файле конфигурации с именем Threads.cfg.

В таблице ниже описаны параметры данного конфигурационного файла.

Таблица 17 — Параметры Threads.cfg

Параметр	Значимость параметра/ перезапуск	Описание
Секция [Physical]		
ThreadsCount	O/R	Количество потоков. Тип — int.
DefaultCPUs	O/R	Номера ядер для обработки потоков. Тип — string.
Threads	Соответствие между потоками и ядрами.	

Пример файла конфигурации:

```

[Physical]
ThreadsCount = 1;
DefaultCPUs = {0;1;2;3;4;5;6;7;};

Threads={
ThreadNumber = 0; CPUs = 0;
}

```

3.4.7 Настройка параметров системы журналирования (trace.cfg)

В конфигурационном файле trace.cfg описываются параметры системы журналирования.

Конфигурация журналов может быть перезагружена средствами динамической команды: ./reload trace.cfg

Примечание — Подробная информация о расшифровке файлов с отладочной информацией предназначена только для служебного пользования. Не рекомендуется менять значения данного файла, так как это может привести к некорректной работе приложения.

В таблице ниже описаны параметры конфигурационного файла.

Таблица 18 — Параметры trace.cfg

Параметр	Значимость параметра/ перезапуск	Описание
Секция [Trace]		
Common	O/R	Набор параметров. Общие настройки системы журналирования. Тип — object. Включает параметры: tracing, dir, no_signal.
tracing	M/R	Активность системы журналирования. Тип — bool. Возможные значения: 0 — система отключена (false); 1 — система активна (true). Значение по умолчанию: 1
dir	O/R	Путь к каталогу, где будут храниться журналы. При необходимости система создаст недостающие каталоги. Тип — string. Если путь начинается с "./", то путь берётся относительно текущего каталога, если с "/", то от корня, иначе — от каталога по умолчанию. Путь может содержать ".." и маску формата времени. Значение по умолчанию: «./logs».
no_signal	O/R	Набор чисел через запятую или строка «all». Сигналы ОС Linux, не перехватываемые системой журналирования. Остальные сигналы система перехватывает и пишет об этом в основные журналы. Тип — list, элементы — int, разделитель — ";", запятая. Возможные значения: "intercept all" — перехватывать все; "all" — вообще не перехватывать сигналы. Значение по умолчанию: "intercept all" — перехватывать все.

Параметр	Значимость параметра/ перезапуск	Описание
Параметры журналов		
log_name	O/R	<p>Имя журнала. Допускаются символы, цифры, знак подчеркивания. Служит идентификатором журнала в системе.</p> <p>Тип — string.</p> <p>Существует несколько зарезервированных имен:</p> <p>trace — журнал «по умолчанию». Содержит параметры: file, mask, level, tee.</p> <p>info — отражает информационные события: создание и течение времени жизни компонентов. Содержит параметры: file, mask, level.</p> <p>warning — журнал предупреждений. Содержит параметры: file, mask, level, tee.</p> <p>error — журнал ошибок.</p> <p>si — журнал системных сообщений.</p> <p>fsm — журнал, отображающий все пересылаемые сообщения.</p> <p>monitor — журнал физического уровня.</p> <p>config_trace — журнал загрузок конфигурационных файлов.</p> <p>config_info — журнал, в котором отображаются информационные сообщения о загрузке конфигурационных файлов.</p> <p>config_warning — журнал ошибок при загрузке конфигурационных файлов.</p> <p>e1_2_rtp — журнал обработки голосовых данных из потока E1.</p> <p>dss_handler — журнал обработки сигнализации DSS-1.</p> <p>isup_handler — журнал обработки подсистемы ISUP.</p> <p>SormDatabase_trace — журнал, содержащий данные по обработке перехватываемых вызовов.</p> <p>voice_manager_trace — журнал работы подсистемы обработки голосового трафика.</p> <p>voice_switch_trace — журнал, содержащий информацию о приеме голосовой информации.</p> <p>pp_trace — журнал обработки перехватываемых пакетов данных.</p>

Параметр	Значимость параметра/ перезапуск	Описание
		<p>pp_info — журнал, в котором отображаются информационные сообщения обработки перехватываемых пакетов данных.</p> <p>pp_warning — журнал ошибок при обработке перехватываемых пакетов данных.</p> <p>Ph_config — журнал загрузки конфигурации аппаратной части.</p> <p>Ph_info — журнал, в котором отображаются информационные сообщения работы аппаратной части.</p> <p>Ph_trace — журнал работы аппаратной части.</p> <p>Ph_warning — журнал ошибок при загрузке конфигурации аппаратной части.</p>
logs	O/R	<p>Набор параметров. Конфигурация журналов.</p> <p>Состоит из параметров следующего формата: log_name = { список_параметров };</p> <p>где log_name — имя_журнала.</p> <p>Тип — object.</p> <p>Описание каждого журнала является опциональным.</p>
file	O/R	<p>Путь к файлу лога.</p> <p>Тип — string.</p> <p>При необходимости недостающие каталоги создаются. Допускается задание пустого имени файла, если level=0, в этом случае запись производится согласно параметру tee. В случае отсутствия этого параметра, запись на диск не производится.</p> <p>Если путь начинается с "./", то путь берётся относительно текущего каталога, если с "/", то от корня, иначе — от каталога по умолчанию. Путь может содержать ".." и маску формата времени.</p> <p>Пример: cdr/%Y/%m/%d/%H_%M_%S.log преобразуется в cdr/2004/07/07/13_54_31.log.</p>
mask	O/R	<p>Маска формата вывода автоматических полей в журнале.</p> <p>Тип — string.</p> <p>Детальное описание смотри в подпункте 3.3.1.2 «Модификаторы mask Ошибка! Источник ссылки не найден.».</p>

Параметр	Значимость параметра/ перезапуск	Описание
level	O/R	<p>Уровень журнала.</p> <p>Тип — int.</p> <p>Специфично для журнала.</p> <p>Примечание — Сообщения с уровнем, большим, чем level, игнорируются.</p>
type	O/R	<p>Тип журнала и дополнительные настройки.</p> <p>Тип — string.</p> <p>Детальное описание смотри в подпункте 3.3.1.4 «Модификаторы type». Ошибка! Источник ссылки не найден.</p>
separator	O/R	<p>Разделитель автоматических полей.</p> <p>Тип — string.</p> <p>Значение по умолчанию: значение параметра common.</p> <p>Примечание — Весь вывод времени (date, time, tick) рассматривается как одно поле.</p>
period	O/R	<p>Период обновления файла лога.</p> <p>Тип — object.</p> <p>Формат: «interval» + «shift», где</p> <p>interval — промежуток времени между соседними обновлениями;</p> <p>shift — первоначальный сдвиг.</p> <p>Детальное описание смотри в подпункте 3.3.1.3 «Модификаторы period».</p> <p>Примечание — Сдвиг не может быть больше длины периода, и в случае некорректного значения игнорируется.</p> <p>Пример: day+3hour — файл обновляется каждый день в 03:00:00.</p>
buffering	O/R	<p>Настройки буферизированной записи.</p> <p>Тип — object.</p> <p>Подпараметры: cluster_size, clusters_in_buffer, overflow_action.</p> <p>Детальное описание смотри в подпункте 3.3.1.1 «Модификаторы buffering».</p>

Параметр	Значимость параметра/ перезапуск	Описание
tee	O/R	<p>Дублирование потока вывода.</p> <p>Тип — string.</p> <p>Возможные значения: stdout, stderr, trace, info, warning или имя любого другого лога.</p> <p>Если перед именем написать «минус», например — «trace», то при дублировании не пишется имя исходного лога.</p> <p>Пример: tee=stdout & stderr & trace & info & warning or any your log.</p>
limit	O/R	<p>Ограничение на максимальное количество строк в файле (записей).</p> <p>Тип — int.</p> <p>Как только достигнут предел строк, лог автоматически открывается заново. При этом не исследуется реальное количество строк в файле на данный момент. Если имя файла зависит от времени, то открывается новый файл, иначе файл обнуляется.</p>

Пример файла конфигурации:

```
[Trace]
common={tracing=1; dir=./logs;}
remote_side = {ip_addr = 192.168.90.2; port_number = 15161;};

logs=
{
  config_trace = {
    file=initialize.log;
    mask=date & time & tick;
    level=10;
  };

  config_info = {
    file=initialize.log;
    mask=date & time & tick;
    level=10;
  };
};
```

```
config_warning = {
    file=initialize.log;
    mask=date & time & tick;
    level=10;
};

e1_2_rtp = {
    file = e1_2_rtp.log;
    mask = date & time & tick;
    level = 10;
}

dss_handler = {
    file = dss_handler.log;
    mask = date & time & tick;
    level = 11;
}

isup_handler = {
    file = isup_handler.log;
    mask = date & time & tick;
    level = 10;
}

SormDatabase_trace = {
    file = sorm_database.log;
    mask = date & time & tick;
    level = 20;
};

voice_manager_trace = {
    file = voice_manager.log;
    mask = date & time & tick & file;
    level = 10;
};

trace = {
    file=trace.log;
```

```
mask=date & time & tick & pid & file;
level=10;
};
error = {
file=error.log;
mask=date & time & tick & pid & file;
level=10;
};
fsm = {
file=fsm.log;
mask=date & time & tick & pid & level & file;
level=10;
};
info = {
file=info.log;
mask=date & time & tick;
level=10;
tee=trace;
};
warning = {
file=warning.log;
mask=date & time & tick & pid & file;
level=9;
tee=trace;
};

si = {
file=si.log;
mask=date & time & tick & pid & file;
level=0;
};

trace_warning = {
file=trace_warning.log;
mask=date & time & tick & pid & file;
level=0;
};
```

```
monitor={
  file = "monitor.log";
  level = "10";
  mask = "date & time & tick & pid & file";
  local_level = "10";
};

Ph_config_info={
  file=Ph_config.log;
  mask=date & time & tick & pid & file;
  level=10;
};

Ph_info = {
  file=Ph_info.log;
  mask=date & time & tick & file;
  level=10;
};

Ph_trace = {
  file=Ph_trace.log;
  mask=date & time & tick & pid & level & file;
  level=10;
};

Ph_warning = {
  file=Ph_warning.log;
  mask=date & time & tick & pid & file;
  level=10;
};

voice_switch_trace = {
  file = voice_switch.log;
  mask = date & time & tick;
  level = 10;
};

PP_trace = {
```

```

file=pp_trace.log;
mask=time & tick & file;
level=100;
};

PP_warning = {
file=pp_warning.log;
mask=time & tick & file;
level=10;
};

PP_info = {
file=pp_info.log;
mask=time & tick & file;
level=10;
};
}

```

3.4.8 Настройка параметров RTP-сессий (VOP.cfg)

В таблице ниже описаны параметры данного конфигурационного файла.

Таблица 19 — Параметры VOP.cfg

Параметр	Значимость параметра/ перезапуск	Описание
Секция [VOP_Defaults]		
EchoEnabled	O/R	<p>Включение эхо-компенсатора.</p> <p>Тип — bool.</p> <p>Возможные значения:</p> <p>0 — не включать;</p> <p>1 — включить.</p> <p>Значение по умолчанию: 1</p>

Параметр	Значимость параметра/ перезапуск	Описание
ConvertDTMF_FromRTP	O/R	<p>Необходимость конвертировать DTMF в акустический сигнал, при приеме из Ethernet RTP-пакета с параметром «payload type» равным 101.</p> <p>Тип — bool.</p> <p>Возможные значения:</p> <p>0 — не конвертировать;</p> <p>1 — конвертировать.</p> <p>Значение по умолчанию: 0</p>
ConvertDTMF_FromPCM	O/R	<p>Необходимость конвертировать DTMF в RTP-пакет с параметром «payload type» равным 101, при приеме из PCM.</p> <p>Примечание — Работает только на платах ИТС.</p> <p>Тип — bool.</p> <p>Возможные значения:</p> <p>0 — не конвертировать;</p> <p>1 — конвертировать.</p> <p>Значение по умолчанию: 0</p>
NotifyDTMF_FromRTP	O/R	<p>Выдавать уведомление на верхний уровень, в случае приема DTMF из Ethernet RTP-пакета с параметром «payload type» равным 101.</p> <p>Примечание — Работает только на платах ИТС.</p> <p>Тип — bool.</p> <p>Возможные значения:</p> <p>0 — не выдавать;</p> <p>1 — выдавать.</p> <p>Значение по умолчанию: 0</p>
NotifyDTMF_FromPCM	O/R	<p>Необходимость выдавать уведомление на верхний уровень, в случае приема DTMF из PCM.</p> <p>Тип — bool.</p> <p>Возможные значения:</p> <p>0 — не выдавать;</p> <p>1 — выдавать.</p> <p>Значение по умолчанию: 1</p>
EnableRTCP	O/R	Необходимость использовать RTCP (Не используется).

Параметр	Значимость параметра/ перезапуск	Описание
BaseIP	M/R	Базовый IP-адрес сетевого интерфейса, который будет использоваться в качестве локального RTP. Тип — int. Значение по умолчанию: 127.0.0.1
BasePort	O/R	Базовый UDP-порт, который будет использоваться в качестве стартового порта локального RTP. Тип — int. Значение по умолчанию: 16384
Секция [G711] Настройки G711		
PacketSize	O/R	Размер пакета на передачу. Тип — int. Значение по умолчанию: 20
JitterSize	O/R	Размер jitter-буфера (в миллисекундах). Тип — int. Значение по умолчанию: 100
Секция [G729] Настройки G729		
PacketSize	O/R	Размер пакета на передачу. Тип — int. Значение по умолчанию: 20
JitterSize	O/R	Размер jitter-буфера (в миллисекундах). Тип — int. Значение по умолчанию: 100

Пример файла конфигурации:

```
[VOP_Defaults]
EchoEnabled = "0";
ConvertDTMF_FromRTP = "0";
ConvertDTMF_FromPCM = "0";
NotifyDTMF_FromRTP = "0";
NotifyDTMF_FromPCM = "0";
EnableRTCP = "1";
```

```

BaseIP = 192.168.90.13;
BasePort = 16384;

[G711]
PacketSize = "20";
JitterSize = "100";
[G729]
PacketSize = "20";
JitterSize = "100";
[G723]
PacketSize = "30";
JitterSize = "100";

```

3.4.9 Настройка физических параметров аппаратного обеспечения PROBE для приема потоков E1

3.4.9.1 Определение используемых файлов конфигурации компонент (config.cfg)

В таблице ниже описаны параметры данного конфигурационного файла.

Таблица 20 — Параметры config.cfg

Параметр	Значимость параметра/ перезапуск	Описание
Секция [Conformity]		
Компонентный адрес. Путь к директории, в которой сохранены настройки параметров драйвера платы и трактов. Тип — string.		
Секция [InitialLoadSequence]		
Последовательность начальной загрузки. Указывается путь к директории, с которой будет осуществляться загрузка системы. Тип — string.		

Пример файла конфигурации:

```

[Conformity]
{"Ph.*";config/component/physical.cfg;}

[InitialLoadSequence]
config/component/physical.cfg;

```

3.4.9.2 Настройка параметров драйвера платы и трактов (physical.cfg)

В таблице ниже описаны параметры данного конфигурационного файла.

Таблица 21 — Параметры physical.cfg

Параметр	Значимость параметра/ перезапуск	Описание
ComponentAddr	O/R	Базовый адрес для работы с платой. Тип — string. Значение по умолчанию: 0xD0000
ComponentType	O/R	Тип компонентного адреса. Тип — string.
Params — составной параметр, содержит набор параметров: «Type», «IRQ», «HardwareRootPath», «ADSP», «BPC». Тип — object.		
Type	O/R	Тип платы. Тип — string. Допустимые значения: ATP, TSP2, TSP3, TSP4, Consul, Consul7.
IRQ	O/R	Номер аппаратного прерывания для платы (устанавливается джампером). Тип — int. Значение по умолчанию: 11
HardwareRootPath	M/R	Путь к драйверам устройств. Тип — string.
ADSP	M/R	Тип программы (прошивки), загружаемой в сигнальный процессор. Тип — string.
BPC	O/R	Использование BPC. Тип — bool. Возможные значения: 0 — не использовать; 1 — использовать. Значение по умолчанию: 0
VoltageType	O/R	Тип напряжения. Тип — int.
Params — составной параметр для создания трактов, содержит параметры Sync, CRC4. Тип — object.		

Параметр	Значимость параметра/ перезапуск	Описание
Sync	O/R	<p>Параметр, описывающий синхронизацию для ИКМ трактов.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> -1(Trunk doesn't exist) — тракт не существует; 0 (No Sync) — от тракта синхронизация не осуществляется; 1 (Lowest) — младший приоритет синхронизации (тракт предпочтителен для синхронизации); 2 (Highest) — наивысший приоритет синхронизации (от тракта идет синхронизация, если нет активных трактов с высоким приоритетом).
CRC4	O/R	<p>Использование CRC4 для данного ИКМ тракта.</p> <p>Тип — bool.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> 0 — не использовать; 1 — использовать. <p>Значение по умолчанию: 0</p>

Пример файла конфигурации:

```
{
ComponentAddr = Ph.Card.0;
ComponentType = Ph.Card;
Params = { Type="consul7";
HardwareRootPath="";
IRQ="4";
IP_Address="192.0.2.2";
};
}
{
ComponentAddr = Ph.Card.0.Trunk.0;
ComponentType = Ph.Trunk;
Params = { Sync="2";
CRC4="0";
};
}
```


4 Техническое обслуживание

4.1 Доступ с использованием команды SSH

Команда SSH служит для установления удаленного сеанса связи.

Для начала работы необходима авторизация пользователя (ввод имени пользователя и пароля). Если номер порта не указан, SSH использует для связи с сервером номер порта по умолчанию (22 порт). Вместо имени сервера может использоваться его IP-адрес.

SSH работает на базе протокола TCP.

При работе в сессии удаленного управления для локализации и решения проблем в основном используются команды, приведённые в таблице ниже.

Таблица 22 — Параметры physical.cfg

Команда	Значение
cd path	переход в каталог path
cd..	переход в родительский каталог
cd /	переход в корневой каталог
ps afx	просмотр запущенных процессов
Pwd	просмотр текущего каталога
ls -l	просмотр содержимого текущего каталога
df -h	информация об использовании места на диске (flash-диск и винчестер)
date	просмотр текущей даты

4.2 Управление работой программного обеспечения

Управление работой системы записи переговоров осуществляется скриптами.

Основными скриптами управления работой программного обеспечения являются: start.sh, stop.sh, status.sh и restart.sh. Использование основных скриптов предоставляет следующие функциональные возможности:

- запуск/остановка системы;
- перезагрузка системы;
- просмотр текущего статуса работы приложения;
- вывод в log-файл информации о текущем состоянии системы.

Основные скрипты, необходимые для управления системой, находятся в корневом каталоге системы: /usr/<путь_до_приложения>/bin.

Ниже приведен перечень и описание основных скриптов:

1. Приложение LI.GW.86:

- **start.sh** — штатный запуск системы;
- **stop.sh** — останавливает систему и все утилиты;
- **restart.sh** — осуществляет перезапуск приложения (аналог "./stop; ./start");

- **version.sh** — вывод информации о версии программного обеспечения;
 - **reload.dump** — перезагрузка настроек записи дампа с потока E1.
2. Приложение Voice-storage:
- **start.sh** — штатный запуск системы;
 - **stop.sh** — останавливает систему и все утилиты;
 - **status.sh** — текущий статус приложения;
 - **restart.sh** — осуществляет перезапуск приложения (аналог `./stop; ./start`).
3. Приложение VOICE-PROBE:
- **start.sh** — штатный запуск системы;
 - **stop.sh** — останавливает систему и все утилиты;
 - **status.sh** — текущий статус приложения;
 - **clear.sh** — очистка лог-файлов.

4.3 Перезапуск системы

При изменении параметров должен производиться перезапуск системы.

Примечание — Путем автоматического перезапуска системы также может осуществляться автоматическая локализация аварий (сбой ПО системы).

Для перезапуска используется команда `./restart.sh` или последовательность команд `./stop.sh;./start.sh`, которая инициализирует загрузку данных и запускает ПО.

Для остановки программного обеспечения используется команда `./stop.sh`.

4.4 Проверка работоспособности

В данном разделе описаны операции, доступные пользователю в веб-интерфейсе приложения.

4.4.1 Проверка работы приложения

Для проверки рабочего состояния приложений следует воспользоваться консолью. Откройте консоль и выполните следующие действия:

1. запустить консоль.
2. ввести команду: `ssh hostname`, где `hostname` — IP-адрес сервера, на котором установлено приложение LI.GW.86, Voice-storage или VOICE-PROBE.
3. ввести регистрационное имя (`login`) и далее пароль.
4. введите соответствующую команду:

```
ps afx| grep access-point-86
```

или

```
ps afx| grep sngi
```

или

```
ps afx| grep -i probe
```

Если в системе появится строка с процессом запуска, то приложение работает корректно.

Если в результате выполнения данной команды не будет выведена никакая информация, то приложение работает не корректно (или остановлено).

Пример выполнения команды:

```
7507 pts/1 S+ 0:00 \_ grep --color=auto sngi
8764 ? Ssl 14518:39 /usr/neos/SNGI/bin/sngi
```

В случае возникновения проблем обратитесь в службу технической поддержки ООО «НЕОС».

4.4.2 Проверка запуска приложения

Для проверки успешного (корректного) запуска приложений следует воспользоваться лог-файлами. Для этого с помощью консоли необходимо перейти в директорию, где хранятся лог-файлы:

- для приложения LI.GW.86: /usr/neos/LI.GW.86/logs;
- для приложения Voice-storage: /usr/neos/SNGI/logs;
- для приложения VOICE-PROBE: /usr/neos/PROBE/logs.

Примечание — Осуществите запуск консоли и пройдите процедуру регистрации, выполнив пункты 1-3 пункта 4.4.1 «Проверка работы приложения».

В системе ведутся следующие лог-файлы:

- для приложения LI.GW.86 — access-point.log.

В access-point.log после успешной загрузки всех компонент и модулей будет выведена строка с информационным сообщением:

```
[2023-02-12 10:41:30,458][INFO ][ID:main][][Access Point 86 v1.0.51 buildNumber1523 349-SNAPSHOT RDC
Protei started ][org.protei.access.point.Main]
```

Если в системе появится строка с соответствующим сообщением, то приложение работает корректно.

Если в результате выполнения данных действий указанная строка не будет выведена в лог-файле access-point.log, то приложение работает не корректно.

- для приложения Voice-storage — out.log.

В out.log после успешной загрузки всех компонент и модулей будет выведена строка с информационным сообщением:

```
voice-storage: success initied
```

Если в системе появится строка с соответствующим сообщением, то приложение работает корректно.

Если в результате выполнения данных действий указанная строка не будет выведена в лог-файле out.log, то приложение работает не корректно.

- для приложения VOICE-PROBE — initialize.log и out.log.

В out.log после успешной загрузки всех компонент и модулей будет выведена строка с информационным сообщением:

```
voice_probe: success initied
```

В initialize.log после успешной загрузки всех компонент и модулей будет выведена строка с информационным сообщением:

```
2023-02-12 13:17:15.340 Config:Application configuration successfully read
```

Если в системе появится строка с соответствующим сообщением, то приложение работает корректно.

Если в результате выполнения данных действий указанная строка не будет выведена в лог-файлах initialize.log и out.log, то приложение работает не корректно.

Примечание — Все ошибки возникающие при запуске приложения и в процессе его работы будут выведены в лог-файлах с соответствующим типом сообщения (WARN или ERROR).

В случае возникновения проблем обратитесь в службу технической поддержки ООО «НЕОС».

4.4.3 Проверка состояния приложения VOICE-PROBE после запуска

Проверить корректность загрузки всех компонент приложения VOICE-PROBE можно отследить по лог-файлу info.log.

Например, компонента, отвечающая за запись данных в БД. При успешной записи данных в лог-файле будет выведена следующая запись:

```
2023-02-12 13:17:15.903 CdrWriter successfully created and inited
```

Если указанная строка не будет найдена в лог-файле, то создание компоненты и подключение к БД было выполнено не корректно.

В случае возникновения проблем обратитесь в службу технической поддержки ООО «НЕОС».

4.4.4 Проверка состояния приложения LI.GW.86 после запуска

Проверить корректность загрузки всех компонент приложения LI.GW.86 можно отследить по лог-файлу info.log.

Например, компонента, отвечающая за проверку настройки портов для подключения ПУ. При успешной записи данных в лог-файле будет выведена следующая запись:

```
[2023-02-12 10:41:40,109][INFO ][ID:server-manager-thr-4][1-0][Started transport server with management port 16118 and data port 16117.][org.protei.access.point.transport.server.MainPuTransportServer]
```

Если указанная строка не будет найдена в лог-файле, то настройка порта была выполнена не корректно.

В случае возникновения проблем обратитесь в службу технической поддержки ООО «НЕОС».

5 Журналы

5.1 Общие принципы организации журналов

Подсистема Tm_Trace позволяет строить древовидную иерархическую структуру журналов системы. Журналы, создаваемые приложением, организованы следующим образом:

- Каждая подсистема пишет свои журналы. Журналы представлены в виде отдельных папок. Каждое событие выводится в отдельной строке. Значения разделены знаком «;». Каждая запись будет начинаться с даты и времени в формате YYYY-MM-DD hh-mm-ss, где YYYY — год, MM — месяц, DD — день, hh — часы, mm — минуты и ss — секунды с начала сбора информации;
- Журналы всех подсистем организованы в виде дерева папок. Перемещаясь по его узлам, можно посмотреть файлы всех отдельных журналов событий;
- Корнем дерева журналов всегда является тип Trace.

Каждая подсистема ведет несколько типов журналов:

1. Тип Trace.

В данном журнале содержится последовательность выполненных команд и значения переменных на данном шаге выполнения программы (трассировку) работы подсистемы. Глубина трассировки управляется параметром level. Сообщения, уровень которых выше, чем настроенный level данного журнала будут игнорироваться системой.

2. Тип Warning.

В данном журнале содержится информация о возникших ошибках в работе трассируемой подсистемы.

3. Тип Info.

В данном журнале содержится информация о событиях, которая представлена для пользователя в упрощенном виде. Данные выводимые системой:

- создание, удаление и модификация ресурсов;
- проход программой некоторых значимых точек.
- состояние внутренних подсистем или окружения приложения.

Помимо подсистемы Tm_Trace часть журналов ведется скриптовой подсистемой, в том числе перенаправленные стандартные потоки вывода приложения.

5.2 Описание журналов приложения LI.GW.86

5.2.1 Журнал работы приложения (access-point.log)

В данном журнале содержится информация о всех событиях приложения, а также предоставляет полную информацию о процессах обработки различных запросов, авариях и ошибках.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример содержимого log-файла:

```
[2023-02-12 12:22:39,654][DEBUG][ID:nioEventLoopGroup-13-1][4-0][Successfully open socket on port 16213
(id=a861eae5).                ][org.protei.access.point.transport.server.R
[2023-02-12 12:22:39,654][DEBUG][ID:nioEventLoopGroup-5-1][3-0][Successfully open socket on port 16182
(id=64fafe38).                ][org.protei.access.point.transport.server.Re
[2023-02-12 12:22:39,668][DEBUG][ID:nioEventLoopGroup-12-1][1-0][Successfully open socket on port 16117
(id=8aad5e37).                ][org.protei.access.point.transport.server.M
[2023-02-12 12:22:39,669][INFO ][ID:server-manager-thr-1][3-0][Started transport server with management port
16182 and data port 16181.    ][org.protei.access.point.transport.server.Reg
[2023-02-12 12:22:39,669][INFO ][ID:server-manager-thr-2][1-0][Started transport server with management port
16118 and data port 16117.    ][org.protei.access.point.transport.server.Mai
[2023-02-12 12:22:39,669][DEBUG][ID:nioEventLoopGroup-11-1][3-0][Successfully open socket on port 16181
(id=c92f8d35).                ][org.protei.access.point.transport.server.R
```

5.2.2 Журнал работы компонент приложения (profiler.log)

В данном журнале содержится наборы характеристик, получаемых во время работы приложения. Собранные данные используются для осуществления профилирования.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример содержимого log-файла:

```
[2023-02-24 00:00:29,241][DEBUG][ID:perfometer-stat-1][state_storage.deleted_files (count:12; total rate:0,000
per/sec; 1 min rate:0,000 per/sec; 5 min rate:0,000 per/sec; 15 min rate:0,000
state_storage.written_files (count:99222; total rate:0,100 per/sec; 1 min rate:0,000 per/sec; 5 min rate:0,000 per/sec;
15 min rate:0,010 per/sec)
agent_socket.send_request (duration:0 ms (min:0 ms; max:1 ms; 75%ile:0 ms; 95%ile:0 ms; 99%ile:0 ms);
count:1389189; total rate:1,400 per/sec)
agent_socket.created_connections (count:1389189; total rate:1,400 per/sec; 1 min rate:1,390 per/sec; 5 min
rate:1,397 per/sec; 15 min rate:1,400 per/sec)
agent_socket.get_response (duration:566 ms (min:1 ms; max:3671 ms; 75%ile:1146 ms; 95%ile:1665 ms; 99%ile:2403
ms); count:1389184; total rate:1,400 per/sec)
][org.protei.perfometer.Perfometer]
```

5.2.3 Журнал работы сенсоров (sensors.journal)

В данном журнале содержится информация о состоянии сенсоров.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример содержимого log-файла:

```
[2023-02-24 12:21:37,028][INFO][1][KeyboardConnectionSensor: Started.    ]
[2023-02-24 12:21:37,193][INFO][1][LogInOutSensor: Started.              ]
[2023-02-24 12:21:37,253][INFO][1][FailedLoginAttemptsSensor: Started.  ]
[2023-02-12 12:21:37,280][ERROR][2][ChassisSensor: Can't start sensor.  ]
```

5.2.4 Журнал процесса работы сенсоров (sensors.log)

В данном журнале содержится информация о процессе сбора и обработки состояния работы сенсоров.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример содержимого log-файла:

```
[2023-02-24 00:00:04,328][DEBUG][ID:sensor-polling-thr-1][Current available memory: 1686484048.][org.protei.access.point.monitoring.alarm.sensor.memory.JVMMemorySensor]
```

5.2.5 Журнал работы компоненты spring (spring.log)

В данном журнале содержится информация о работе компоненты spring, входящий в состав приложения.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример содержимого log-файла:

```
[2023-02-24 12:04:09,654][DEBUG][ID:main][Creating shared instance of singleton bean 'monitoringConfigurationContext' ][org.springframework.beans.factory.support.DefaultListableBeanFactory]
[2023-02-24 12:04:09,654][TRACE][ID:main][Creating instance of bean 'monitoringConfigurationContext' ][org.springframework.beans.factory.support.DefaultListableBeanFactory]
[2023-02-24 12:04:09,654][TRACE][ID:main][Eagerly caching bean 'monitoringConfigurationContext' to allow for resolving potential circular references][org.springframework.beans.factory.support.DefaultListableBeanFactory]
[2023-02-24 12:04:09,655][TRACE][ID:main][Finished creating instance of bean 'monitoringConfigurationContext' ][org.springframework.beans.factory.support.DefaultListableBeanFactory]
```

5.2.6 Системный журнал (system.log)

Системный журнал, содержащий сведения о состоянии приложения в определенный момент времени.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример содержимого log-файла:

```
Сб фев 24 12:22:14 MSK 2021 send stop query to pid 38819
Сб фев 24 12:22:29 MSK 2021 started
Process has started with PID = 39821
```

5.3 Описание журналов приложения Voice-storage

5.3.1 Базовые журналы приложения

5.3.1.1 Основной журнал приложения (trace.log)

В данном журнале содержится информация о всех событиях приложения, а также предоставляет полную информацию о процессах обработки различных запросов.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример содержимого log-файла:

```
2023-02-24 14:48:07.824 004911 VoiceMixer.cpp(68) VoiceMixer: Regular tracks : track1.startTime = 1605688986722
and track2.startTime = 1605688986177
2023-02-24 14:48:07.824 004911 VoiceMixer.cpp(89) VoiceMixer: Track1 1605688986722 - 29200 1460
2023-02-24 14:48:07.824 004911 VoiceMixer.cpp(90) VoiceMixer: Track2 1605688986177 - 29900 1494
2023-02-24 14:48:07.824 004911 VoiceMixer.cpp(98) VoiceMixer: Total length: 29900
```

5.3.1.2 Журнал предупреждений (warning.log)

В данном журнале содержится информация о некритичных ошибках в работе приложения. Данные, выводимые в журнале, позволяют пользователю в короткие сроки определить причины некорректной работы приложения.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

5.3.1.3 Журнал мониторинга (info.log)

В данном журнале фиксируется информация о параметрах RTP-потоков.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

5.3.2 Стандартный вывод и журнал скриптовой подсистемы

Стандартные потоки вывода в системе — потоки процесса, имеющие номер, зарезервированный для выполнения некоторых «стандартных» функций.

Стандартный вывод (stdout)

Поток stdout зарезервирован для вывода данных, фиксирующих версии запускаемых подсистем, процесс их запуска и работы приложения. С помощью глобального параметра OUT_FILE, задаваемого в конфигурационном файле usr/neos/SNGI/bin/scripts.sh, осуществляется корректный вывод значений.

Журнал, в котором хранятся все данные stdout, размещается в out.log.

Пример конфигурации:

```
OUT_FILE=./logs/out.log
```

Пример записи:

```
voice-storage: start device on port 1
voice-storage: start device on port - 1, device name - af on rtp_in, with mac - 76:B6:15:92:A1:12
voice-storage: success inited
voice-storage: will start

af on rtp_in:
rx packets: 1257
rx errors : 1
tx packets: 537
tx errors : 0
rx packets drop : 0
total pkt in receiver_route : 1256
```

```
total pkt in worker_route : 537
ipv6 packets drop : 0
```

Стандартный вывод ошибок (stderr)

Поток stderr зарезервирован для вывода диагностических и отладочных сообщений в текстовом формате для пользователя. С помощью потока stderr фиксируются критичные ошибки в работе приложения. При корректной работе должен оставаться пустым.

Вывод значений осуществляется с помощью глобального параметра ERR_FILE, задаваемого в конфигурационном файле usr/neos/SNGI/bin/scripts.sh.

Каталог, в котором хранятся все данные журналов размещается в err.log.

Пример конфигурации:

```
ERR_FILE=./logs/err.log
```

5.3.3 Журнал переданных RTP-поток (rtp_sender.log)

В данном журнале фиксируются данные о передаче RTP-поток к другим приложениям.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

5.3.4 Журнал ошибок передачи RTP-поток (rtp_sender_warning.log)

В данном журнале фиксируется информация об ошибках при передаче RTP-поток к другим приложениям. Данные, выводимые в журнале, позволяют пользователю в короткие сроки определить ошибки при передаче RTP-поток.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

5.4 Описание журналов приложения VOICE-PROBE

5.4.1 Базовые журналы приложения

Основными журналами приложения являются: trace.log, info.log и warning.log. Детальное описание базовых журналов аналогично приведённому выше для приложения Voice-storage.

Назначение лог-файлов out.log и err.log, осуществляющих стандартный вывод данных и диагностических сообщений, приведено в пункте 5.3.2 «Стандартный вывод и журнал скриптовой подсистемы»

5.4.2 Журнал обработки сигнализации DSS1 (dss1_handler.log)

В данном журнале фиксируется информация об обработке информации по протоколу DSS1.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример содержимого log-файла:

```
2023-02-24 16:59:10.441 Dss1Decoder [0xb8e2184]: New message was received from: Ph.Card.0.Trunk.1.TSL.16
2023-02-24 16:59:10.441 Dss1Decoder [0xb8e2184]: Decoding message header
2023-02-24 16:59:10.441 Dss1Decoder [0xb8e2184]: | PD: 8
2023-02-24 16:59:10.442 Dss1Decoder [0xb8e2184]: | CR: 2
2023-02-24 16:59:10.442 Dss1Decoder [0xb8e2184]: | MT: 5 (SETUP)
2023-02-24 16:59:10.442 Dss1Decoder [0xb8e2184]: Decoding message payload
2023-02-24 16:59:10.442 Dss1Handler [0xb86673c]: Handling message SETUP
2023-02-24 16:59:10.442 Dss1Decoder [0xb8e2184]: Message was successfully decoded
```

5.4.3 Журнал обработки сигнализации ISUP (isup_handler.log)

В данном журнале фиксируется информация об обработке информации по протоколу ISUP.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример содержимого log-файла:

```
2023-02-24 17:37:12.755 ISUP::Listener [0xb95f154]: ISUP::Listener [0xb95f154] was successfully created
2023-02-24 17:37:12.755 ISUP::Manager [0xb95f594]: ISUP::Manager [0xb95f594] was successfully created
```

5.4.4 Журнал обработки голосового трафика (voice_manager.log)

В данном журнале фиксируется информация об обработке голосового трафика и его передачи на приложение SNGI.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример содержимого log-файла:

```
2023-02-24 14:58:24.072 [[LII.VM]] New call: #"dss:7:49" has content
2023-02-24 14:58:24.072 [[LII.VM]] Sending RTP_CONNECT_REQ(20423) for Ph.Card.2.Trunk.4.TSL.23
2023-02-24 14:58:24.073 [[LII.VM]] Sending RTP_CONNECT_REQ(20523) for Ph.Card.2.Trunk.5.TSL.23
2023-02-24 14:58:24.082 [[LII.VM]] Received RTP_CONNECT_CONF(20423) for Local RTP: [10.1.81.5:16400]
2023-02-24 14:58:24.082 [[LII.VM]] Updating RtpContent:#4579882361361006690(5.81.1.10:16400) for call:
#"dss:7:49" in CallJournal
2023-02-24 14:58:24.093 [[LII.VM]] Received RTP_CONNECT_CONF(20523) for Local RTP: [10.1.81.5:16402]
2023-02-24 14:58:24.093 [[LII.VM]] Updating RtpContent:#4579882361361006691(5.81.1.10:16402) for call:
#"dss:7:49" in CallJournal
```

5.4.5 Журнал приема голосовой информации (voice_switch.log)

В данном журнале содержится информация о проключении голосового трафика от источника.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

5.4.6 Журнал обработки голосовых данных из потока E1 (e1_2_rtp.log)

В данном журнале содержится информация о преобразовании голосовых данных из потока E1 в RTP-потоки.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример содержимого log-файла:

```
2023-02-24 14:58:05.433 [[LII.E1_RTP.SL.1]] VOP channel create confirm: obj: Ph.Card.2.ITC.19.TSL.1 local port: 16386
2023-02-24 14:58:05.432 [[LII.E1_RTP.SL.0]] VOP channel create confirm: obj: Ph.Card.2.ITC.19.TSL.0 local port: 16384
2023-02-24 14:58:05.439 [[LII.E1_RTP.SL.0]] Connect success: Ph.Card.2.Trunk.2.TSL.25 ---> Ph.Card.2.ITC.19.TSL.0
2023-02-24 14:58:05.441 [[LII.E1_RTP.SL.1]] Connect success: Ph.Card.2.Trunk.3.TSL.25 ---> Ph.Card.2.ITC.19.TSL.1
```

5.4.7 Журнал обработки перехватываемых вызовов (sorm_database.log)

В данном журнале содержится информация об отобранных перехваченных вызовах и их запись в БД.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

Пример содержимого log-файла:

```
2023-02-24 16:59:10.471 CallJournal: Add requested by 'Dss1Listener' with Call #"dss:2" "5191" -> "3191" [DEBUG]
2023-02-24 16:59:10.471 CallJournal: Running transaction Add Call from 'Dss1Listener' [DEBUG]
2023-02-24 16:59:10.471 CallJournal: : Adding Call: #"dss:2" "5191" -> "3191"
```

5.4.8 Журнал загрузки приложения (initialize.log)

В данном журнале содержится информация о загрузке всех компонент приложения.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

5.4.9 Журнал конфигурации аппаратного обеспечения (Ph_config.log)

В данном журнале содержится информация о загруженной конфигурации аппаратного обеспечения.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

5.4.10 Журнал состояния аппаратного обеспечения (Ph_info.log)

В данном журнале содержится информация о состоянии аппаратного обеспечения.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

5.4.11 Журнал работы аппаратного обеспечения (Ph_trace.log)

В данном журнале содержится детальная информация о работе аппаратного обеспечения.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.

5.4.12 Журнал ошибок аппаратного обеспечения (Ph_warning.log)

В данном журнале содержится информация об ошибках, возникающих при работе аппаратного обеспечения.

Параметры журнала настраиваются в файле конфигурации Trace.cfg.